In [1]:
```
using Pkg
Pkg.activate("..")
Pkg.instantiate()
Pkg.update()
```

```
 Activating project at `~/logic-and-machine-learning`
   Updating registry at `~/.julia/registries/General.toml`
   Updating git-repo `https://github.com/aclai-lab/ManyExpertDecisionTree
s.jl`
   Updating git-repo `https://github.com/aclai-lab/SoleReasoners.jl#embed
ding`
  No Changes to `~/logic-and-machine-learning/Project.toml`
  No Changes to `~/logic-and-machine-learning/Manifest.toml`
```

In [2]:
```
using Random

Random.seed!(1235)
```

Out[2]:
```
TaskLocalRNG()
```

Identifying academically vulnerable learners in first-year science programmes

Identifying academically vulnerable learners in first-year science programmes at a South African higher-education institution

In [3]:
```
using ARFFFiles
using DataFrames

data = ARFFFiles.load(
    DataFrame,
    joinpath("..", "datasets", "academically-vulnerable-learners.arff")
)

describe(data)
```

Out[3]: 45×7 DataFrame *20 rows omitted*

Row	variable	mean	min	median	max	nmissing
	Symbol	Union…	Any	Union…	Any	Int64
1	Risk Status		Medium Risk		High Risk	0
2	YearStarted	2012.62	2008.0	2012.59	2018.0	0
3	PlanCode		SB000DFLT		SPROGSOC90	0
4	PlanDescription		SB000 Default		Science Occ Studies (UG)	0
5	Majors		MATH		PALP	0
6	APS Code		STATNSC		AIN11	0
7	APS Value	39.6162	0.0	41.4386	68.0	94
8	Streamline		Unknown Streamline		Earth Science Degree	0
9	Homeprovince		GA		KZ	0
10	Homecountry		ZAF		ZMB	0
11	AgeatFirstYear	24.722	17.0	24.8654	40.0	0
12	SchoolQuintile		Q2		NaN	0
13	isRuralorUrban		RURAL		NaN	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
34	Civil	90.0	90.0	90.0	90.0	799
35	BusinessEconomics	59.8053	0.0	73.0	88.0	755
36	ReligiousStudies	37.2	0.0	0.0	100.0	795
37	PhysicsChem	56.3617	0.0	65.5511	100.0	119
38	InternationalRelations	63.5714	0.0	66.0	90.0	793
39	Geography	64.2156	0.0	72.0	98.0	613
40	Hospitality	78.0	78.0	78.0	78.0	799
41	History	63.1071	0.0	79.0	91.0	772
42	Politics	NaN				800
43	LifeSciences	57.4927	0.0	71.865	97.0	218

Row	variable	mean	min	median	max	nmissing
	Symbol	Union…	Any	Union…	Any	Int64
44	ForiegnSubject	NaN				800
45	SportScience	70.0	70.0	70.0	70.0	799

Oh no! Some attributes have maaany missing values!!!

In [4]:

```
attributes_with_missings = Vector{Tuple{String, Int}}()

for attribute_name in names(data)
    n_missings = count(x -> ismissing(x), data[:, attribute_name])

    if n_missings > 0
        push!(attributes_with_missings, (attribute_name, n_missings))
    end
end

attxmiss = sort!(attributes_with_missings, by = x -> x[2], rev = false)
```

Out[4]:

```
33-element Vector{Tuple{String, Int64}}:
 ("APS Value", 94)
 ("MathematicsMatricMajor", 98)
 ("PhysicsChem", 119)
 ("AdditionalLanguage", 137)
 ("LifeOrientation", 213)
 ("LifeSciences", 218)
 ("NBTAL", 443)
 ("NBTQL", 448)
 ("NBTMA", 450)
 ("EnglishFirstLang", 472)
 ("EnglishFirstAdditional", 607)
 ("Geography", 613)
 ("AdditionalMathematics", 668)
 ⋮
 ("InternationalRelations", 793)
 ("ReligiousStudies", 795)
 ("Music", 797)
 ("MathematicsMatricLit", 798)
 ("Mechanical", 798)
 ("Civil", 799)
 ("Hospitality", 799)
 ("SportScience", 799)
 ("Statistics", 800)
 ("OverallAggregate", 800)
 ("Politics", 800)
 ("ForiegnSubject", 800)
```

We have some preprocessing to do!

Let's start by dropping the columns with more missing values.

In [5]:

```
# we want to drop these features
colstodrop = [feature for (feature, nmisses) in attxmiss[11:end]]
```

Out[5]:

```
23-element Vector{String}:
 "EnglishFirstAdditional"
 "Geography"
 "AdditionalMathematics"
 "Accounting"
 "ComputerStudies"
 "BusinessEconomics"
 "History"
 "Agriculture"
 "Electrical"
 "ArtDesign"
 "CraftSpeechDrama"
 "InternationalRelations"
 "ReligiousStudies"
 "Music"
 "MathematicsMatricLit"
 "Mechanical"
 "Civil"
 "Hospitality"
 "SportScience"
 "Statistics"
 "OverallAggregate"
 "Politics"
 "ForiegnSubject"
```

In [6]:

```julia
[select!(data, Not(Symbol(col))) for col in colstodrop];    # remember th
describe(data)
```

Out[6]: 22×7 DataFrame

Row	variable	mean	min	median	max	nmissi
	Symbol	Union…	Any	Union…	Any	Int64
1	Risk Status		Medium Risk		High Risk	
2	YearStarted	2012.62	2008.0	2012.59	2018.0	
3	PlanCode		SB000DFLT		SPROGSOC90	
4	PlanDescription		SB000 Default		Science Occ Studies (UG)	
5	Majors		MATH		PALP	
6	APS Code		STATNSC		AIN11	
7	APS Value	39.6162	0.0	41.4386	68.0	
8	Streamline		Unknown Streamline		Earth Science Degree	
9	Homeprovince		GA		KZ	
10	Homecountry		ZAF		ZMB	
11	AgeatFirstYear	24.722	17.0	24.8654	40.0	
12	SchoolQuintile		Q2		NaN	
13	isRuralorUrban		RURAL		NaN	
14	LifeOrientation	64.0089	0.0	79.0245	100.0	2
15	MathematicsMatricMajor	58.0737	0.0	69.0	100.0	
16	EnglishFirstLang	64.6558	0.0	71.0	95.4928	4
17	NBTAL	61.6059	31.0	63.2377	87.0	4
18	NBTMA	50.7234	25.0	49.0	94.0	4
19	NBTQL	55.0162	24.5085	54.0442	94.0	4
20	AdditionalLanguage	64.1048	0.0	71.3761	98.0	1
21	PhysicsChem	56.3617	0.0	65.5511	100.0	1
22	LifeSciences	57.4927	0.0	71.865	97.0	2

We still have some missings!

Let's remove rows with missing values.

In [7]:
```
using Impute

data_nomissing = Impute.filter(data; dims=:rows)
```

Out[7]: 118×22 DataFrame *93 rows omitted*

Row	Risk Status	YearStarted	PlanCode	PlanDescription	Majors	APS Code	AP Va
	Cat…	Float64	Cat…	Cat…	Cat…	Cat…	Flo
1	No Risk	2011.0	SMAJADES10	Advanced Earth Sciences	GEOG	STATNSC	
2	Medium Risk	2017.0	SMAJGEN00	General	COMS	AIN12	
3	Low Risk	2009.77	SMAJASTR10	Astronomy Astrophysics	COMS	STATNSC	40
4	Low Risk	2013.0	SMAJEEC10	Ecology, Environment and Conservation	APES	AIN12	
5	No Risk	2012.83	SMAJAEBI10	Applied & Experimental Physiology	PHSL	STATNSC	44
6	No Risk	2016.0	SMAJACSI10	Actuarial Science	MATH	AIN12	
7	Low Risk	2015.0	SMAJAPES10	Animal, Plant and Environmental Sciences	APES	AIN12	
8	No Risk	2012.01	SMAJBIO10	Biochemistry and Cell Biology	MCBG	STATNSC	40
9	Medium Risk	2013.0	SMAJCOMM10	Computer Science	MATH	AIN12	
10	No Risk	2010.0	SMAJAEBI10	Applied & Experimental Physiology	MCBG	STATNSC	
11	Medium Risk	2011.0	SB000DFLT	SB000 Default	ECON	STATNSC	
12	Low Risk	2011.0	SMAJHPHS10	Human Physiology	PHSL	STATNSC	
13	High Risk	2013.0	SMAJGENE10	Genetics & Development Biology	MCBG	AIN12	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	
107	No Risk	2016.0	SMAJGEN00	General	CHEM	AIN12	

Row	Risk Status	YearStarted	PlanCode	PlanDescription	Majors	APS Code	AP Va
	Cat...	Float64	Cat...	Cat...	Cat...	Cat...	Flo
108	Medium Risk	2010.0	SB000DFLT	SB000 Default	BIOL	STATNSC	
109	Low Risk	2010.43	SMAJASTR10	Astronomy Astrophysics	MATH	AIN12	53
110	No Risk	2014.03	SMAJAEBI10	Applied & Experimental Physiology	PHSL	AIN12	40
111	Low Risk	2012.0	SMAJBIO10	Biochemistry and Cell Biology	MCBG	STATNSC	
112	High Risk	2014.99	SMAJBIOL10	Biological Sciences	MATH	AIN12	41
113	No Risk	2012.0	SMAJGENE10	Genetics & Development Biology	MCBG	STATNSC	
114	Medium Risk	2009.0	SB000DFLT	SB000 Default	MATH	STATNSC	
115	Medium Risk	2018.0	SMAGEN00	General	MATH	AIN12	
116	Low Risk	2010.0	SMAJCAMS10	Computational and Applied Mathematics	MATH	STATNSC	
117	No Risk	2016.0	SMAJECOS10	Economic Science	MATH	AIN12	
118	No Risk	2016.0	SMAJGENE10	Genetics & Development Biology	MCBG	AIN12	

In [8]:
```
using MLJ

schema(data_nomissing)
```

Out[8]:

```
┌─────────────────────────┬─────────────────┬─────────────────────────────
──────┐
│ names                   │ scitypes        │ types
│
├─────────────────────────┼─────────────────┼─────────────────────────────
──────┤
│ Risk Status             │ Multiclass{4}   │ CategoricalValue{String, UIn
t32} │
│ YearStarted             │ Continuous      │ Float64
│
│ PlanCode                │ Multiclass{60}  │ CategoricalValue{String, UIn
t32} │
│ PlanDescription         │ Multiclass{49}  │ CategoricalValue{String, UIn
t32} │
│ Majors                  │ Multiclass{29}  │ CategoricalValue{String, UIn
t32} │
│ APS Code                │ Multiclass{8}   │ CategoricalValue{String, UIn
t32} │
│ APS Value               │ Continuous      │ Float64
│
│ Streamline              │ Multiclass{5}   │ CategoricalValue{String, UIn
t32} │
│ Homeprovince            │ Multiclass{14}  │ CategoricalValue{String, UIn
t32} │
│ Homecountry             │ Multiclass{25}  │ CategoricalValue{String, UIn
t32} │
│ AgeatFirstYear          │ Continuous      │ Float64
│
│ SchoolQuintile          │ Multiclass{6}   │ CategoricalValue{String, UIn
t32} │
│ isRuralorUrban          │ Multiclass{3}   │ CategoricalValue{String, UIn
t32} │
│ LifeOrientation         │ Continuous      │ Float64
│
│ MathematicsMatricMajor  │ Continuous      │ Float64
│
│ EnglishFirstLang        │ Continuous      │ Float64
│
│ NBTAL                   │ Continuous      │ Float64
│
│ NBTMA                   │ Continuous      │ Float64
│
│ NBTQL                   │ Continuous      │ Float64
│
│ AdditionalLanguage      │ Continuous      │ Float64
│
│ PhysicsChem             │ Continuous      │ Float64
│
│ LifeSciences            │ Continuous      │ Float64
│
└─────────────────────────┴─────────────────┴─────────────────────────────
──────┘
```

Let's see which kind of models we could use...

In [9]:

```
y, X = unpack(data_nomissing, ==(Symbol("Risk Status")))

models(matching(X,y))
```

```
Out[9]: 7-element Vector{NamedTuple{(:name, :package_name, :is_supervised, :abstract_type, :constructor, :deep_properties, :docstring, :fit_data_scitype, :human_name, :hyperparameter_ranges, :hyperparameter_types, :hyperparameters, :implemented_methods, :inverse_transform_scitype, :is_pure_julia, :is_wrapper, :iteration_parameter, :load_path, :package_license, :package_url, :package_uuid, :predict_scitype, :prediction_type, :reporting_operations, :reports_feature_importances, :supports_class_weights, :supports_online, :supports_training_losses, :supports_weights, :target_in_fit, :transform_scitype, :input_scitype, :target_scitype, :output_scitype)}}:
 (name = CatBoostClassifier, package_name = CatBoost, ... )
 (name = ConstantClassifier, package_name = MLJModels, ... )
 (name = DecisionTreeClassifier, package_name = BetaML, ... )
 (name = DeterministicConstantClassifier, package_name = MLJModels, ... )
 (name = EvoTreeClassifier, package_name = EvoTrees, ... )
 (name = NeuralNetworkClassifier, package_name = MLJFlux, ... )
 (name = RandomForestClassifier, package_name = BetaML, ... )
```

Too bad! Most models don't work with categorical values out of the box...

This includes the `DecisionTreeClassifier` from `DecisionTree.jl` !

Hence, we first need to encode these values as numerical values.

One possibility is to convert the type of the associated features from `Multiclass` to `Continuous` or `OrderedFactor` .

```
In [10]: data_preprocessed = coerce(data_nomissing, "Risk Status"=>OrderedFactor)
data_preprocessed = coerce(data_preprocessed, Multiclass=>Continuous)

schema(data_preprocessed)
```

Out[10]:

```
┌──────────────────────────┬────────────────┬────────────────────────────
────────
│ names                    │ scitypes       │ types
…
├──────────────────────────┼────────────────┼────────────────────────────
────────
│ Risk Status              │ OrderedFactor{4} │ CategoricalValue{String, U
Int32} …
│ YearStarted              │ Continuous     │ Float64
…
│ PlanCode                 │ Continuous     │ Float64
…
│ PlanDescription          │ Continuous     │ Float64
…
│ Majors                   │ Continuous     │ Float64
…
│ APS Code                 │ Continuous     │ Float64
…
│ APS Value                │ Continuous     │ Float64
…
│ Streamline               │ Continuous     │ Float64
…
│ Homeprovince             │ Continuous     │ Float64
…
│ Homecountry              │ Continuous     │ Float64
…
│ AgeatFirstYear           │ Continuous     │ Float64
…
│ SchoolQuintile           │ Continuous     │ Float64
…
│ isRuralorUrban           │ Continuous     │ Float64
…
│ LifeOrientation          │ Continuous     │ Float64
…
│ MathematicsMatricMajor   │ Continuous     │ Float64
…
│ EnglishFirstLang         │ Continuous     │ Float64
…
│ NBTAL                    │ Continuous     │ Float64
…
│ NBTMA                    │ Continuous     │ Float64
…
│ NBTQL                    │ Continuous     │ Float64
…
│ AdditionalLanguage       │ Continuous     │ Float64
…
│ PhysicsChem              │ Continuous     │ Float64
…
│ LifeSciences             │ Continuous     │ Float64
…
└──────────────────────────┴────────────────┴────────────────────────────
────────
```

Let's have a look at the data...

In [11]:

```
y, X = unpack(data_preprocessed, ==(Symbol("Risk Status")))
```

Out[11]:

```
(CategoricalArrays.CategoricalValue{String, UInt32}["No Risk", "Medium R
isk", "Low Risk", "Low Risk", "No Risk", "No Risk", "Low Risk", "No Risk
", "Medium Risk", "No Risk"  …  "Low Risk", "No Risk", "Low Risk", "High
Risk", "No Risk", "Medium Risk", "Medium Risk", "Low Risk", "No Risk", "
No Risk"], 118×21 DataFrame
 Row │ YearStarted  PlanCode  PlanDescription  Majors   APS Code  APS Va
lue  S …
     │ Float64      Float64   Float64          Float64  Float64   Float6
4    F …
─────┼──────────────────────────────────────────────────────────────────
────────
   1 │     2011.0       18.0             17.0      9.0       1.0    46.0
…
   2 │     2017.0       10.0             10.0     11.0       2.0    40.0
   3 │     2009.77       2.0              2.0     11.0       1.0    40.8
845
   4 │     2013.0        7.0              7.0      6.0       2.0    39.0
   5 │     2012.83       3.0              3.0      3.0       1.0    44.0
344    …
   6 │     2016.0       15.0             14.0      1.0       2.0    50.0
   7 │     2015.0       33.0             29.0      6.0       2.0    42.0
   8 │     2012.01       5.0              5.0      4.0       1.0    40.5
945
   9 │     2013.0       27.0             18.0      1.0       2.0    39.0
…
  10 │     2010.0        3.0              3.0      4.0       1.0    49.0
  11 │     2011.0        1.0              1.0     15.0       1.0    36.0
  ⋮  │      ⋮          ⋮             ⋮           ⋮         ⋮         ⋮
⋱
 109 │     2010.43       2.0              2.0      1.0       2.0    53.4
871
 110 │     2014.03       3.0              3.0      3.0       2.0    40.7
176    …
 111 │     2012.0        5.0              5.0      4.0       1.0    40.0
 112 │     2014.99       4.0              4.0      1.0       2.0    41.3
076
 113 │     2012.0       17.0             16.0      4.0       1.0    39.0
 114 │     2009.0        1.0              1.0      1.0       1.0    42.0
…
 115 │     2018.0       13.0             10.0      1.0       2.0    48.0
 116 │     2010.0       22.0             20.0      1.0       1.0    46.0
 117 │     2016.0        9.0              9.0      1.0       2.0    43.0
 118 │     2016.0       17.0             16.0      4.0       2.0    43.0
…
                                                 15 columns and 97 rows
omitted)
```

Great! We can now use a `DecisionTreeClassifier` like in our example!

In [12]:

```
models(matching(X,y))
```

Out[12]:
```
54-element Vector{NamedTuple{(:name, :package_name, :is_supervised, :abs
tract_type, :constructor, :deep_properties, :docstring, :fit_data_scityp
e, :human_name, :hyperparameter_ranges, :hyperparameter_types, :hyperpar
ameters, :implemented_methods, :inverse_transform_scitype, :is_pure_juli
a, :is_wrapper, :iteration_parameter, :load_path, :package_license, :pac
kage_url, :package_uuid, :predict_scitype, :prediction_type, :reporting_
operations, :reports_feature_importances, :supports_class_weights, :supp
orts_online, :supports_training_losses, :supports_weights, :target_in_fi
t, :transform_scitype, :input_scitype, :target_scitype, :output_scityp
e)}}:
 (name = AdaBoostClassifier, package_name = MLJScikitLearnInterface, ...
)
 (name = AdaBoostStumpClassifier, package_name = DecisionTree, ... )
 (name = BaggingClassifier, package_name = MLJScikitLearnInterface, ...
)
 (name = BayesianLDA, package_name = MLJScikitLearnInterface, ... )
 (name = BayesianLDA, package_name = MultivariateStats, ... )
 (name = BayesianQDA, package_name = MLJScikitLearnInterface, ... )
 (name = BayesianSubspaceLDA, package_name = MultivariateStats, ... )
 (name = CatBoostClassifier, package_name = CatBoost, ... )
 (name = ConstantClassifier, package_name = MLJModels, ... )
 (name = DecisionTreeClassifier, package_name = BetaML, ... )
 (name = DecisionTreeClassifier, package_name = DecisionTree, ... )
 (name = DeterministicConstantClassifier, package_name = MLJModels, ...
)
 (name = DummyClassifier, package_name = MLJScikitLearnInterface, ... )
 ⋮
 (name = RandomForestClassifier, package_name = MLJScikitLearnInterface,
... )
 (name = RidgeCVClassifier, package_name = MLJScikitLearnInterface, ...
)
 (name = RidgeClassifier, package_name = MLJScikitLearnInterface, ... )
 (name = SGDClassifier, package_name = MLJScikitLearnInterface, ... )
 (name = SVC, package_name = LIBSVM, ... )
 (name = SVMClassifier, package_name = MLJScikitLearnInterface, ... )
 (name = SVMLinearClassifier, package_name = MLJScikitLearnInterface, ..
. )
 (name = SVMNuClassifier, package_name = MLJScikitLearnInterface, ... )
 (name = StableForestClassifier, package_name = SIRUS, ... )
 (name = StableRulesClassifier, package_name = SIRUS, ... )
 (name = SubspaceLDA, package_name = MultivariateStats, ... )
 (name = XGBoostClassifier, package_name = XGBoost, ... )
```

Let's first choose a random sample from our dataset: we will use it later to evaluate our model.

In [13]:
```
y, X = unpack(data_preprocessed, ==(Symbol("Risk Status")))

(X_train, X_test), (y_train, y_test) = partition(
    (X, y),
    0.8,
    rng=13,
    shuffle=true,
    multi=true
);
```

Let's try to work following the pipeline we learned this week!

In [14]:
```
try
    DecisionTreeClassifier = @load DecisionTreeClassifier pkg=DecisionTre
catch
    println("The DecisionTreeClassifier symbol has already been imported.
end
```

```
[ Info: For silent loading, specify `verbosity=0`.
import MLJDecisionTreeInterface ✔
```

Out[14]:
```
MLJDecisionTreeInterface.DecisionTreeClassifier
```

In [15]:
```
model = MLJDecisionTreeInterface.DecisionTreeClassifier()
```

Out[15]:
```
DecisionTreeClassifier(
  max_depth = -1,
  min_samples_leaf = 1,
  min_samples_split = 2,
  min_purity_increase = 0.0,
  n_subfeatures = 0,
  post_prune = false,
  merge_purity_threshold = 1.0,
  display_depth = 5,
  feature_importance = :impurity,
  rng = TaskLocalRNG())
```

In [16]:
```
mach = machine(model, X_train, y_train)
```

Out[16]:
```
untrained Machine; caches model-specific representations of data
  model: DecisionTreeClassifier(max_depth = -1, …)
  args:
    1:  Source @399 ⏎ Table{AbstractVector{Continuous}}
    2:  Source @546 ⏎ AbstractVector{OrderedFactor{4}}
```

In [17]:
```
fit!(mach)
```

```
[ Info: Training machine(DecisionTreeClassifier(max_depth = -1, …), …).
```

Out[17]:
```
trained Machine; caches model-specific representations of data
  model: DecisionTreeClassifier(max_depth = -1, …)
  args:
    1:  Source @399 ⏎ Table{AbstractVector{Continuous}}
    2:  Source @546 ⏎ AbstractVector{OrderedFactor{4}}
```

In [18]:
```
🌱 = fitted_params(mach).tree   # \:seedling:
```

Out[18]:

```
YearStarted < 2016.0
├─ Majors < 1.5
│  ├─ APS Value < 42.05
│  │  ├─ LifeOrientation < 87.76
│  │  │  ├─ Medium Risk (8/8)
│  │  │  └─ High Risk (1/1)
│  │  └─ AdditionalLanguage < 74.95
│  │     ├─ High Risk (5/5)
│  │     └─ AgeatFirstYear < 27.6
│  │        ├─ Low Risk (7/7)
│  │        └─ Medium Risk (1/1)
│  └─ Majors < 5.0
│     ├─ PlanCode < 6.5
│     │  ├─ Homeprovince < 1.5
│     │  │  ├─ YearStarted < 2010.0
│     │  │  │  ⋮
│     │  │  │
│     │  │  └─ AgeatFirstYear < 22.5
│     │  │     ⋮
│     │  │
│     │  └─ YearStarted < 2013.0
│     │     ├─ Homeprovince < 1.5
│     │     │  ⋮
│     │     │
│     │     └─ LifeOrientation < 23.71
│     │        ⋮
│     │
│     └─ PhysicsChem < 75.0
│        ├─ Majors < 7.5
│        │  ├─ Low Risk (5/5)
│        │  └─ PlanDescription < 14.0
│        │     ⋮
│        │
│        └─ MathematicsMatricMajor < 87.5
│           ├─ MathematicsMatricMajor < 85.0
│           │  ⋮
│           │
│           └─ Low Risk (5/5)
└─ Medium Risk (17/17)
```

Let's evaluate performance!

In [19]:

```
y_predict_probabilities = predict(mach, X_test)
y_predict = mode.(y_predict_probabilities)
cm = confusion_matrix(y_predict, y_test)
```

Out[19]:

	Ground Truth			
Predicted	Medium R…	Low Risk	No Risk	High Risk
Medium R…	12	0	0	2
Low Risk	0	4	2	0
No Risk	0	0	4	0
High Risk	0	0	0	0

In [20]:
```julia
accuracy(cm)
```

Out[20]:
```
0.8333333333333334
```

Let's extract logical rules!

In [21]:
```julia
using SoleModels

🌲 = solemodel(🌱)  # \:evergreen_tree:
```

```
WARNING: using SoleModels.models in module Main conflicts with an existing
identifier.
```

Out[21]:
```
▣ ([YearStarted] < 2016.5)
├✔ ([Majors] < 1.5)
│ ├✔ ([APS Value] < 42.0546825)
│ │ ├✔ ([LifeOrientation] < 87.761518)
│ │ │ ├✔ Medium Risk
│ │ │ └✘ High Risk
│ │ └✘ ([AdditionalLanguage] < 74.950632)
│ │   ├✔ High Risk
│ │   └✘ ([AgeatFirstYear] < 27.5967815)
│ │     ├✔ Low Risk
│ │     └✘ Medium Risk
│ └✘ ([Majors] < 5.0)
│   ├✔ ([PlanCode] < 6.5)
│   │ ├✔ ([Homeprovince] < 1.5)
│   │ │ ├✔ ([YearStarted] < 2009.5)
│   │ │ │ ├✔ Low Risk
│   │ │ │ └✘ No Risk
│   │ │ └✘ ([AgeatFirstYear] < 22.5)
│   │ │   ├✔ Medium Risk
│   │ │   └✘ ([AdditionalLanguage] < 34.0)
│   │ │     ├✔ No Risk
│   │ │     └✘ Low Risk
│   │ └✘ ([YearStarted] < 2013.2996785)
│   │   ├✔ ([Homeprovince] < 1.5)
│   │   │ ├✔ Low Risk
│   │   │ └✘ ([YearStarted] < 2012.5)
│   │   │   ├✔ No Risk
│   │   │   └✘ High Risk
│   │   └✘ ([LifeOrientation] < 23.71375)
│   │     ├✔ Low Risk
│   │     └✘ No Risk
│   └✘ ([PhysicsChem] < 75.0)
│     ├✔ ([Majors] < 7.5)
│     │ ├✔ Low Risk
│     │ └✘ ([PlanDescription] < 14.0)
│     │   ├✔ Medium Risk
│     │   └✘ Low Risk
│     └✘ ([MathematicsMatricMajor] < 87.5)
│       ├✔ ([MathematicsMatricMajor] < 85.0)
│       │ ├✔ No Risk
│       │ └✘ Medium Risk
│       └✘ Low Risk
└✘ Medium Risk
```

In [22]:
```julia
listrules(🌲)
```

```
Out[22]: 22-element Vector{ClassificationRule{String}}:
 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] < 1.5)) ∧ (([APS Value] < 4
2.0546825)) ∧ (([LifeOrientation] < 87.761518))  ↣  Medium Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] < 1.5)) ∧ (([APS Value] < 4
2.0546825)) ∧ (([LifeOrientation] ≥ 87.761518))  ↣  High Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] < 1.5)) ∧ (([APS Value] ≥ 4
2.0546825)) ∧ (([AdditionalLanguage] < 74.950632))  ↣  High Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] < 1.5)) ∧ (([APS Value] ≥ 4
2.0546825)) ∧ (([AdditionalLanguage] ≥ 74.950632)) ∧ (([AgeatFirstYear]
< 27.5967815))  ↣  Low Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] < 1.5)) ∧ (([APS Value] ≥ 4
2.0546825)) ∧ (([AdditionalLanguage] ≥ 74.950632)) ∧ (([AgeatFirstYear]
≥ 27.5967815))  ↣  Medium Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] < 6.5)) ∧ (([Homeprovince] < 1.5)) ∧ (([YearStarted] < 20
09.5))  ↣  Low Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] < 6.5)) ∧ (([Homeprovince] < 1.5)) ∧ (([YearStarted] ≥ 20
09.5))  ↣  No Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] < 6.5)) ∧ (([Homeprovince] ≥ 1.5)) ∧ (([AgeatFirstYear] <
22.5))  ↣  Medium Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] < 6.5)) ∧ (([Homeprovince] ≥ 1.5)) ∧ (([AgeatFirstYear] ≥
22.5)) ∧ (([AdditionalLanguage] < 34.0))  ↣  No Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] < 6.5)) ∧ (([Homeprovince] ≥ 1.5)) ∧ (([AgeatFirstYear] ≥
22.5)) ∧ (([AdditionalLanguage] ≥ 34.0))  ↣  Low Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] ≥ 6.5)) ∧ (([YearStarted] < 2013.2996785)) ∧ (([Homeprovi
nce] < 1.5))  ↣  Low Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] ≥ 6.5)) ∧ (([YearStarted] < 2013.2996785)) ∧ (([Homeprovi
nce] ≥ 1.5)) ∧ (([YearStarted] < 2012.5))  ↣  No Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] ≥ 6.5)) ∧ (([YearStarted] < 2013.2996785)) ∧ (([Homeprovi
nce] ≥ 1.5)) ∧ (([YearStarted] ≥ 2012.5))  ↣  High Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] ≥ 6.5)) ∧ (([YearStarted] ≥ 2013.2996785)) ∧ (([LifeOrien
tation] < 23.71375))  ↣  Low Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] < 5.0))
∧ (([PlanCode] ≥ 6.5)) ∧ (([YearStarted] ≥ 2013.2996785)) ∧ (([LifeOrien
tation] ≥ 23.71375))  ↣  No Risk

 ▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] ≥ 5.0))
∧ (([PhysicsChem] < 75.0)) ∧ (([Majors] < 7.5))  ↣  Low Risk
```

▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] ≥ 5.0)) ∧ (([PhysicsChem] < 75.0)) ∧ (([Majors] ≥ 7.5)) ∧ (([PlanDescription] < 14.0)) ↣ Medium Risk

▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] ≥ 5.0)) ∧ (([PhysicsChem] < 75.0)) ∧ (([Majors] ≥ 7.5)) ∧ (([PlanDescription] ≥ 14.0)) ↣ Low Risk

▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] ≥ 5.0)) ∧ (([PhysicsChem] ≥ 75.0)) ∧ (([MathematicsMatricMajor] < 87.5)) ∧ (([MathematicsMatricMajor] < 85.0)) ↣ No Risk

▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] ≥ 5.0)) ∧ (([PhysicsChem] ≥ 75.0)) ∧ (([MathematicsMatricMajor] < 87.5)) ∧ (([MathematicsMatricMajor] ≥ 85.0)) ↣ Medium Risk

▣ (([YearStarted] < 2016.5)) ∧ (([Majors] ≥ 1.5)) ∧ (([Majors] ≥ 5.0)) ∧ (([PhysicsChem] ≥ 75.0)) ∧ (([MathematicsMatricMajor] ≥ 87.5)) ↣ Low Risk

▣ ([YearStarted] ≥ 2016.5) ↣ Medium Risk

Let's evaluate each formula (or logical rule) separately.

In [23]:
```
apply!(🌲, X_test, y_test);
metricstable(
    🌲;
    normalize = true,
    metrics_kwargs = (;
        additional_metrics = (;
            height = r->SoleLogics.height(antecedent(r))
        )
    )
)
```

Antecedent	Consequent	ninstances	ncovered	coverage	confidence	lift	natoms	height
([YearStarted] < 2016.5) ∧ ([Majors] < 1.5) ∧ ([APS Value] < 42.0546825) ∧ ([LifeOrientation] < 87.761518)	Medium Risk	24	9	0.375	0.777778	1.55556	4	3
([YearStarted] < 2016.5) ∧ ([Majors] < 1.5) ∧ ([APS Value] < 42.0546825) ∧ ([LifeOrientation] ≥ 87.761518)	High Risk	24	0	0.0	NaN	NaN	4	3
([YearStarted] < 2016.5) ∧ ([Majors] < 1.5) ∧ ([APS Value] ≥ 42.0546825) ∧ ([AdditionalLanguage] < 74.950632)	High Risk	24	0	0.0	NaN	NaN	4	3
([YearStarted] < 2016.5) ∧ ([Majors] < 1.5) ∧ ([APS Value] ≥ 42.0546825) ∧ ([AdditionalLanguage] ≥ 74.950632) ∧ ([AgeatFirstYear] < 27.5967815)	Low Risk	24	2	0.0833333	0.0	0.0	5	4
([YearStarted] < 2016.5) ∧ ([Majors] < 1.5) ∧ ([APS Value] ≥ 42.0546825) ∧ ([AdditionalLanguage] ≥ 74.950632) ∧ ([AgeatFirstYear] ≥ 27.5967815)	Medium Risk	24	0	0.0	NaN	NaN	5	4
([YearStarted] < 2009.5) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] < 6.5) ∧ ([Homeprovince] < 1.5)	Low Risk	24	0	0.0	NaN	NaN	4	3
([YearStarted] ∈ [2009.5,2016.5)) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] < 6.5) ∧ ([Homeprovince] < 1.5)	No Risk	24	4	0.166667	1.0	4.0	4	3
([YearStarted] < 2016.5) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] < 6.5) ∧ ([Homeprovince] ≥ 1.5) ∧ ([AgeatFirstYear] < 22.5)	Medium Risk	24	0	0.0	NaN	NaN	5	4
([YearStarted] < 2016.5) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] < 6.5) ∧ ([Homeprovince] ≥ 1.5) ∧ ([AgeatFirstYear] ≥ 22.5) ∧ ([AdditionalLanguage] < 34.0)	No Risk	24	0	0.0	NaN	NaN	6	5
([YearStarted] < 2016.5) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] < 6.5) ∧ ([Homeprovince] ≥ 1.5) ∧ ([AgeatFirstYear] ≥ 22.5) ∧ ([AdditionalLanguage] ≥ 34.0)	Low Risk	24	0	0.0	NaN	NaN	6	5
([YearStarted] < 2013.2996785) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] ≥ 6.5) ∧ ([Homeprovince] < 1.5)	Low Risk	24	1	0.0416667	1.0	6.0	4	3
([YearStarted] < 2012.5) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] ≥ 6.5) ∧ ([Homeprovince] ≥ 1.5)	No Risk	24	0	0.0	NaN	NaN	4	3
([YearStarted] ∈ [2012.5,20								

```
13.2996785)) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] ≥ 6.5) ∧ ([Homeprovinc
e] ≥ 1.5) │    High Risk │          24 │          0 │        0.0 │          NaN │
NaN │       4 │       3 │
│                                       ([YearStarted] ∈ [2013.2996785,201
6.5)) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] ≥ 6.5) ∧ ([LifeOrientation] <
23.71375) │     Low Risk │          24 │          0 │        0.0 │          NaN │
NaN │       4 │       3 │
│                                       ([YearStarted] ∈ [2013.2996785,201
6.5)) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] ≥ 6.5) ∧ ([LifeOrientation] ≥
23.71375) │      No Risk │          24 │          0 │        0.0 │          NaN │
NaN │       4 │       3 │
│
([YearStarted] < 2016.5) ∧ ([Majors] ∈ [5.0,7.5)) ∧ ([PhysicsChem] < 75.0)
│     Low Risk │          24 │          1 │ 0.0416667 │          1.0 │       6.0 │
3 │       2 │
│                                                            ([YearStarted]
< 2016.5) ∧ ([Majors] ≥ 7.5) ∧ ([PhysicsChem] < 75.0) ∧ ([PlanDescription]
< 14.0) │ Medium Risk │          24 │          0 │        0.0 │          NaN │
NaN │       4 │       3 │
│                                                            ([YearStarted]
< 2016.5) ∧ ([Majors] ≥ 7.5) ∧ ([PhysicsChem] < 75.0) ∧ ([PlanDescription]
≥ 14.0) │     Low Risk │          24 │          2 │ 0.0833333 │          1.0 │
6.0 │       4 │       3 │
│                                                       ([YearStarted] < 201
6.5) ∧ ([Majors] ≥ 5.0) ∧ ([PhysicsChem] ≥ 75.0) ∧ ([MathematicsMatricMajo
r] < 85.0) │     No Risk │          24 │          0 │        0.0 │          NaN
│      NaN │       4 │       3 │
│                                                ([YearStarted] < 2016.5) ∧
([Majors] ≥ 5.0) ∧ ([PhysicsChem] ≥ 75.0) ∧ ([MathematicsMatricMajor] ∈ [8
5.0,87.5)) │ Medium Risk │          24 │          0 │        0.0 │          NaN
│      NaN │       4 │       3 │
│                                                       ([YearStarted] < 201
6.5) ∧ ([Majors] ≥ 5.0) ∧ ([PhysicsChem] ≥ 75.0) ∧ ([MathematicsMatricMajo
r] ≥ 87.5) │     Low Risk │          24 │          0 │        0.0 │          NaN
│      NaN │       4 │       3 │
│
[YearStarted] ≥ 2016.5 │ Medium Risk │          24 │          5 │   0.208333 │
1.0 │     2.0 │       1 │       0 │
└──────────────────────────────────────────────────────────────────────────
───────────────────────────────────────────────────────────────────────────
──────────┴─────────────┴────────────┴──────────┴────────────┴────────────┴
─────────┴────────┴────────┘
```

Let's summarize our model joining rules associated with the same class!

In [24]:
```
metricstable(joinrules(🌲; min_ncovered = 1, normalize = true))
```

Antecedent	Consequent	ninstances	ncovered	coverage	confidence	lift	natoms
(([YearStarted] < 2016.5) ∧ ([Majors] < 1.5) ∧ ([APS Value] < 42.0546825) ∧ ([LifeOrientation] < 87.761518)) ∨ ([YearStarted] ≥ 2016.5)	Medium Risk	48	14	0.291667	0.857143	1.71429	5
(([YearStarted] < 2016.5) ∧ ([Majors] < 1.5) ∧ ([APS Value] ≥ 42.0546825) ∧ ([AdditionalLanguage] ≥ 74.950632) ∧ ([AgeatFirstYear] < 27.5967815)) ∨ (([YearStarted] < 2013.2996785) ∧ ([Majors] ∈ [1.5,5.0)) ∧ ([PlanCode] ≥ 6.5) ∧ ([Homeprovince] < 1.5)) ∨ (([YearStarted] < 2016.5) ∧ ([Majors] ∈ [5.0,7.5)) ∧ ([PhysicsChem] < 75.0)) ∨ (([YearStarted] < 2016.5) ∧ ([Majors] ≥ 7.5) ∧ ([PhysicsChem] < 75.0) ∧ ([PlanDescription] ≥ 14.0))	Low Risk	96	6	0.0625	0.666667	4.0	16
[YearStarted] ∈ [2009.5,2016.5) ∧ [Majors] ∈ [1.5,5.0) ∧ [PlanCode] < 6.5 ∧ [Homeprovince] < 1.5	No Risk	24	4	0.166667	1.0	4.0	4

Let's now try to learn a random forest.

In [25]:
```
try
    RandomForestClassifier = @load RandomForestClassifier pkg=DecisionTre
catch
    println("The RandomForestClassifier symbol has already been imported.
end
```

```
[ Info: For silent loading, specify `verbosity=0`.
import MLJDecisionTreeInterface ✔
```

Out[25]:
```
MLJDecisionTreeInterface.RandomForestClassifier
```

In [26]:
```
forest = MLJDecisionTreeInterface.RandomForestClassifier(n_trees=10)
```

Out[26]:
```
RandomForestClassifier(
  max_depth = -1,
  min_samples_leaf = 1,
  min_samples_split = 2,
  min_purity_increase = 0.0,
  n_subfeatures = -1,
  n_trees = 10,
  sampling_fraction = 0.7,
  feature_importance = :impurity,
  rng = TaskLocalRNG())
```

In [27]:
```
forestmach = machine(forest, X_train, y_train)
```

Out[27]:
```
untrained Machine; caches model-specific representations of data
  model: RandomForestClassifier(max_depth = -1, …)
  args:
    1:  Source @568 ⏎ Table{AbstractVector{Continuous}}
    2:  Source @586 ⏎ AbstractVector{OrderedFactor{4}}
```

In [28]:
```
MLJ.fit!(forestmach, verbosity=0)
```

Out[28]:
```
trained Machine; caches model-specific representations of data
  model: RandomForestClassifier(max_depth = -1, …)
  args:
    1:  Source @568 ⏎ Table{AbstractVector{Continuous}}
    2:  Source @586 ⏎ AbstractVector{OrderedFactor{4}}
```

In [29]:
```
🌱🌱🌱 = fitted_params(forestmach).forest   # \:seedling:
```

Out[29]:
```
Ensemble of Decision Trees
Trees:      10
Avg Leaves: 17.3
Avg Depth:  7.4
```

Let's evaluate its performance.

In [30]:
```
y_predict_probabilities = MLJ.predict(forestmach, X_test)
y_predict = mode.(y_predict_probabilities)
cm = confusion_matrix(y_predict, y_test)
```

Out[30]:

	Ground Truth			
Predicted	Medium R…	Low Risk	No Risk	High Risk
Medium R…	11	0	2	0
Low Risk	1	4	0	0
No Risk	0	0	4	0
High Risk	0	0	0	2

In [31]:
```
accuracy(cm)
```

Out[31]:
```
0.875
```

Let's extract logical rules!

In [32]:

```
🌲🌲🌲 = solemodel(🌱🌱🌱)  # \:evergreen_tree:
```

```
Out[32]: ▣ Ensemble{UInt32} of 10 models of type Branch{UInt32}
         ├[1/10]┐ (V1 < 2015.015294)
         │       ├✔ (V2 < 1.5)
         │       │ ├✔ 1 : (ninstances = 5, ncovered = 5, confidence = 1.0, lift =
         1.0)
         │       │ └✘ (V19 < 45.665659500000004)
         │       │   ├✔ 3 : (ninstances = 5, ncovered = 5, confidence = 1.0, lift
         = 1.0)
         │       │   └✘ (V19 < 71.94951499999999)
         │       │     ├✔ (V20 < 79.614036)
         │       │     │ ├✔ (V2 < 7.0)
         │       │     │ │ ├✔ (V3 < 5.5)
         │       │     │ │ │ ├✔ 3 : (ninstances = 1, ncovered = 1, confidence = 1.
         0, lift = 1.0)
         │       │     │ │ │ └✘ 4 : (ninstances = 5, ncovered = 5, confidence = 1.
         0, lift = 1.0)
         │       │     │ │ └✘ (V7 < 3.5)
         │       │     │ │   ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.
         0, lift = 1.0)
         │       │     │ │   └✘ (V15 < 68.5)
         │       │     │ │     ├✔ 2 : (ninstances = 2, ncovered = 2, confidence =
         1.0, lift = 1.0)
         │       │     │ │     └✘ 3 : (ninstances = 1, ncovered = 1, confidence =
         1.0, lift = 1.0)
         │       │     │ └✘ 2 : (ninstances = 5, ncovered = 5, confidence = 1.0, l
         ift = 1.0)
         │       │     └✘ (V5 < 1.5)
         │       │       ├✔ (V4 < 6.0)
         │       │       │ ├✔ (V13 < 74.0)
         │       │       │ │ ├✔ 3 : (ninstances = 4, ncovered = 4, confidence = 1.
         0, lift = 1.0)
         │       │       │ │ └✘ (V19 < 83.5)
         │       │       │ │   ├✔ 2 : (ninstances = 2, ncovered = 2, confidence =
         1.0, lift = 1.0)
         │       │       │ │   └✘ 3 : (ninstances = 2, ncovered = 2, confidence =
         1.0, lift = 1.0)
         │       │       │ └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0,
         lift = 1.0)
         │       │       └✘ 2 : (ninstances = 8, ncovered = 8, confidence = 1.0, l
         ift = 1.0)
         │       └✘ (V14 < 76.5)
         │         ├✔ (V1 < 2017.0)
         │         │ ├✔ (V4 < 2.0)
         │         │ │ ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
         t = 1.0)
         │         │ │ └✘ 3 : (ninstances = 14, ncovered = 14, confidence = 1.0, l
         ift = 1.0)
         │         │ └✘ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, lift
         = 1.0)
         │         └✘ 1 : (ninstances = 5, ncovered = 5, confidence = 1.0, lift =
         1.0)
         ├[2/10]┐ (V19 < 37.026329000000004)
         │       ├✔ 3 : (ninstances = 7, ncovered = 7, confidence = 1.0, lift =
         1.0)
         │       └✘ (V1 < 2015.5)
         │         ├✔ (V7 < 3.5)
         │         │ ├✔ (V11 < 3.5)
         │         │ │ ├✔ (V15 < 71.2623505)
         │         │ │ │ ├✔ (V4 < 4.0)
         │         │ │ │ │ ├✔ (V5 < 1.5)
```

```
│        │ │ │ │ │ ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.
0, lift = 1.0)
│        │ │ │ │ │ └✘ 4 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│        │ │ │ │ └✘ 2 : (ninstances = 3, ncovered = 3, confidence = 1.0,
lift = 1.0)
│        │ │ │ └✘ (V21 < 76.811504)
│        │ │ │   ├✔ (V13 < 45.986777)
│        │ │ │   │ ├✔ 2 : (ninstances = 3, ncovered = 3, confidence = 1.
0, lift = 1.0)
│        │ │ │   │ └✘ 4 : (ninstances = 2, ncovered = 2, confidence = 1.
0, lift = 1.0)
│        │ │ │   └✘ 2 : (ninstances = 8, ncovered = 8, confidence = 1.0,
lift = 1.0)
│        │ │ └✘ (V2 < 10.0)
│        │ │   ├✔ (V16 < 76.63369900000001)
│        │ │   │ ├✔ 4 : (ninstances = 3, ncovered = 3, confidence = 1.0,
lift = 1.0)
│        │ │   │ └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│        │ │   └✘ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
│        │ └✘ (V20 < 66.0)
│        │   ├✔ (V2 < 6.0)
│        │   │ ├✔ (V19 < 80.41465199999999)
│        │   │ │ ├✔ 3 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
│        │   │ │ └✘ (V10 < 28.0)
│        │   │ │   ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│        │   │ │   └✘ 2 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│        │   │ └✘ 2 : (ninstances = 5, ncovered = 5, confidence = 1.0, l
ift = 1.0)
│        │   └✘ 3 : (ninstances = 5, ncovered = 5, confidence = 1.0, lif
t = 1.0)
│        └✘ (V3 < 13.0)
│          ├✔ (V15 < 64.5)
│          │ ├✔ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
│          │ └✘ 1 : (ninstances = 10, ncovered = 10, confidence = 1.0, l
ift = 1.0)
│          └✘ (V2 < 33.5)
│            ├✔ 3 : (ninstances = 4, ncovered = 4, confidence = 1.0, lif
t = 1.0)
│            └✘ (V10 < 20.5)
│              ├✔ 1 : (ninstances = 3, ncovered = 3, confidence = 1.0, l
ift = 1.0)
│              └✘ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
├[3/10]┐ (V4 < 1.5)
│      ├✔ (V3 < 8.0)
│      │ ├✔ (V3 < 3.0)
│      │ │ ├✔ (V2 < 1.5)
│      │ │ │ ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, lif
t = 1.0)
│      │ │ │ └✘ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
│      │ │ └✘ 4 : (ninstances = 10, ncovered = 10, confidence = 1.0, lif
t = 1.0)
```

```
│      │ └✘ 1 : (ninstances = 14, ncovered = 14, confidence = 1.0, lift
= 1.0)
│      └✘ (V1 < 2016.5)
│        ├✔ (V4 < 7.0)
│        │ ├✔ (V21 < 87.5)
│        │ │ ├✔ (V14 < 63.0)
│        │ │ │ ├✔ 3 : (ninstances = 9, ncovered = 9, confidence = 1.0, l
ift = 1.0)
│        │ │ │ └✘ (V20 < 57.0)
│        │ │ │   ├✔ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│        │ │ │   └✘ (V10 < 24.0)
│        │ │ │     ├✔ 3 : (ninstances = 4, ncovered = 4, confidence = 1.
0, lift = 1.0)
│        │ │ │     └✘ (V14 < 68.2470955)
│        │ │ │       ├✔ 3 : (ninstances = 1, ncovered = 1, confidence =
1.0, lift = 1.0)
│        │ │ │       └✘ (V13 < 76.0)
│        │ │ │         ├✔ 3 : (ninstances = 1, ncovered = 1, confidence
= 1.0, lift = 1.0)
│        │ │ │         └✘ 2 : (ninstances = 2, ncovered = 2, confidence
= 1.0, lift = 1.0)
│        │ │ └✘ 2 : (ninstances = 3, ncovered = 3, confidence = 1.0, lif
t = 1.0)
│        │ └✘ (V13 < 80.5)
│        │   ├✔ (V10 < 25.0)
│        │   │ ├✔ (V15 < 76.5)
│        │   │ │ ├✔ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│        │   │ │ └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│        │   │ └✘ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
│        │   └✘ (V4 < 10.0)
│        │     ├✔ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
│        │     └✘ 2 : (ninstances = 3, ncovered = 3, confidence = 1.0, l
ift = 1.0)
│        └✘ 1 : (ninstances = 9, ncovered = 9, confidence = 1.0, lift =
1.0)
├[4/10]┐ (V21 < 73.8124525)
│      ├✔ (V19 < 77.0)
│      │ ├✔ (V1 < 2011.9126425)
│      │ │ ├✔ (V2 < 6.5)
│      │ │ │ ├✔ (V4 < 2.0)
│      │ │ │ │ ├✔ 4 : (ninstances = 3, ncovered = 3, confidence = 1.0, l
ift = 1.0)
│      │ │ │ │ └✘ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
│      │ │ │ └✘ (V7 < 3.0)
│      │ │ │   ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
│      │ │ │   └✘ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
│      │ │ └✘ (V3 < 17.0)
│      │ │   ├✔ 3 : (ninstances = 12, ncovered = 12, confidence = 1.0, l
ift = 1.0)
│      │ │   └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
│      │ └✘ (V19 < 79.247703)
```

```
│      │   ├✔ 2 : (ninstances = 4, ncovered = 4, confidence = 1.0, lift
= 1.0)
│      │   └✘ 4 : (ninstances = 3, ncovered = 3, confidence = 1.0, lift
= 1.0)
│      └✘ (V14 < 90.0)
│        ├✔ (V19 < 71.318906)
│        │ ├✔ (V7 < 2.0)
│        │ │ ├✔ 3 : (ninstances = 4, ncovered = 4, confidence = 1.0, lif
t = 1.0)
│        │ │ └✘ (V13 < 77.5)
│        │ │   ├✔ 2 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
│        │ │   └✘ 3 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
│        │ └✘ (V21 < 82.5)
│        │   ├✔ 1 : (ninstances = 14, ncovered = 14, confidence = 1.0, l
ift = 1.0)
│        │   └✘ (V3 < 25.0)
│        │     ├✔ 1 : (ninstances = 7, ncovered = 7, confidence = 1.0, l
ift = 1.0)
│        │     └✘ (V2 < 46.0)
│        │       ├✔ 2 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
│        │       └✘ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│        └✘ 2 : (ninstances = 6, ncovered = 6, confidence = 1.0, lift =
1.0)
├[5/10]┐ (V10 < 20.5)
│      ├✔ 1 : (ninstances = 11, ncovered = 11, confidence = 1.0, lift =
1.0)
│      └✘ (V1 < 2011.0241065)
│        ├✔ (V20 < 62.9324955)
│        │ ├✔ 1 : (ninstances = 5, ncovered = 5, confidence = 1.0, lift
= 1.0)
│        │ └✘ (V21 < 81.5)
│        │   ├✔ 2 : (ninstances = 5, ncovered = 5, confidence = 1.0, lif
t = 1.0)
│        │   └✘ (V15 < 75.29259400000001)
│        │     ├✔ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
│        │     └✘ (V13 < 82.5)
│        │       ├✔ 3 : (ninstances = 4, ncovered = 4, confidence = 1.0,
lift = 1.0)
│        │       └✘ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
│        └✘ (V4 < 9.0)
│          ├✔ (V7 < 2.5)
│          │ ├✔ 3 : (ninstances = 15, ncovered = 15, confidence = 1.0, l
ift = 1.0)
│          │ └✘ (V2 < 7.0)
│          │   ├✔ (V4 < 2.0)
│          │   │ ├✔ 4 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
│          │   │ └✘ 3 : (ninstances = 7, ncovered = 7, confidence = 1.0,
lift = 1.0)
│          │   └✘ (V10 < 24.578322)
│          │     ├✔ (V16 < 78.0)
│          │     │ ├✔ 3 : (ninstances = 5, ncovered = 5, confidence = 1.
0, lift = 1.0)
│          │     │ └✘ 2 : (ninstances = 2, ncovered = 2, confidence = 1.
```

```
0, lift = 1.0)
│          │     └✘ (V3 < 12.0)
│          │       ├✔ 2 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│          │       └✘ 4 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│          └✘ (V15 < 76.5)
│            ├✔ 2 : (ninstances = 3, ncovered = 3, confidence = 1.0, lif
t = 1.0)
│            └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
├[6/10]┐ (V10 < 20.5)
│      ├✔ 1 : (ninstances = 18, ncovered = 18, confidence = 1.0, lift =
1.0)
│      └✘ (V4 < 2.0)
│        ├✔ (V6 < 41.0546825)
│        │ ├✔ 1 : (ninstances = 5, ncovered = 5, confidence = 1.0, lift
= 1.0)
│        │ └✘ (V7 < 2.5)
│        │   ├✔ 2 : (ninstances = 4, ncovered = 4, confidence = 1.0, lif
t = 1.0)
│        │   └✘ 4 : (ninstances = 5, ncovered = 5, confidence = 1.0, lif
t = 1.0)
│        └✘ (V1 < 2015.4700990000001)
│          ├✔ (V2 < 6.0)
│          │ ├✔ (V20 < 3.678195)
│          │ │ ├✔ (V6 < 43.0)
│          │ │ │ ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│          │ │ │ └✘ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│          │ │ └✘ 3 : (ninstances = 5, ncovered = 5, confidence = 1.0, l
ift = 1.0)
│          │ └✘ (V19 < 71.0)
│          │   ├✔ (V11 < 2.0)
│          │   │ ├✔ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│          │   │ └✘ (V10 < 25.0)
│          │   │   ├✔ 2 : (ninstances = 3, ncovered = 3, confidence = 1.
0, lift = 1.0)
│          │   │   └✘ 3 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│          │   └✘ (V17 < 39.0)
│          │     ├✔ (V21 < 41.0)
│          │     │ ├✔ 4 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│          │     │ └✘ 2 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│          │     └✘ 2 : (ninstances = 11, ncovered = 11, confidence = 1.
0, lift = 1.0)
│          └✘ 3 : (ninstances = 8, ncovered = 8, confidence = 1.0, lift
= 1.0)
├[7/10]┐ (V1 < 2016.5)
│      ├✔ (V6 < 39.3047615)
│      │ ├✔ (V17 < 26.0)
│      │ │ ├✔ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0, lift
= 1.0)
│      │ │ └✘ (V19 < 77.0)
│      │ │   ├✔ (V16 < 70.5)
│      │ │   │ ├✔ (V17 < 30.5)
```

```
│      │ │   │ │ ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
│      │ │   │ │ └✘ (V2 < 4.0)
│      │ │   │ │   ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
│      │ │   │ │   └✘ 2 : (ninstances = 5, ncovered = 5, confidence = 1.
0, lift = 1.0)
│      │ │   │ └✘ 1 : (ninstances = 3, ncovered = 3, confidence = 1.0, l
ift = 1.0)
│      │ │   └✘ 2 : (ninstances = 6, ncovered = 6, confidence = 1.0, lif
t = 1.0)
│      │ └✘ (V4 < 10.0)
│      │   ├✔ (V13 < 82.5)
│      │   │ ├✔ (V2 < 20.5)
│      │   │ │ ├✔ (V2 < 5.5)
│      │   │ │ │ ├✔ 3 : (ninstances = 10, ncovered = 10, confidence = 1.
0, lift = 1.0)
│      │   │ │ │ └✘ (V2 < 13.0)
│      │   │ │ │   ├✔ (V3 < 8.0)
│      │   │ │ │   │ ├✔ 4 : (ninstances = 1, ncovered = 1, confidence =
1.0, lift = 1.0)
│      │   │ │ │   │ └✘ 1 : (ninstances = 1, ncovered = 1, confidence =
1.0, lift = 1.0)
│      │   │ │ │   └✘ (V3 < 17.5)
│      │   │ │ │     ├✔ 3 : (ninstances = 5, ncovered = 5, confidence =
1.0, lift = 1.0)
│      │   │ │ │     └✘ 1 : (ninstances = 1, ncovered = 1, confidence =
1.0, lift = 1.0)
│      │   │ │ └✘ (V10 < 22.0)
│      │   │ │   ├✔ 3 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
│      │   │ │   └✘ 2 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
│      │   │ └✘ (V3 < 3.5)
│      │   │   ├✔ 2 : (ninstances = 3, ncovered = 3, confidence = 1.0, l
ift = 1.0)
│      │   │   └✘ 1 : (ninstances = 3, ncovered = 3, confidence = 1.0, l
ift = 1.0)
│      │   └✘ 2 : (ninstances = 7, ncovered = 7, confidence = 1.0, lift
= 1.0)
│      └✘ 1 : (ninstances = 13, ncovered = 13, confidence = 1.0, lift =
1.0)
├[8/10]┐ (V18 < 53.0)
│      ├✔ (V7 < 3.5)
│      │ ├✔ (V16 < 59.5)
│      │ │ ├✔ (V3 < 8.0)
│      │ │ │ ├✔ (V20 < 75.114036)
│      │ │ │ │ ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
│      │ │ │ │ └✘ 4 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
│      │ │ │ └✘ 1 : (ninstances = 3, ncovered = 3, confidence = 1.0, lif
t = 1.0)
│      │ │ └✘ (V18 < 42.8810585)
│      │ │   ├✔ 4 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
│      │ │   └✘ (V14 < 69.774136)
│      │ │     ├✔ (V15 < 64.42533)
│      │ │     │ ├✔ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
```

```
|      | |     | └✘ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
|      | |     └✘ 3 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
|      | └✘ (V6 < 41.0)
|      |   ├✔ (V19 < 76.5)
|      |   | ├✔ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
|      |   | └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
|      |   └✘ (V7 < 4.5)
|      |     ├✔ 3 : (ninstances = 7, ncovered = 7, confidence = 1.0, lif
t = 1.0)
|      |     └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
|      └✘ (V14 < 85.316749)
|        ├✔ (V13 < 81.5)
|        | ├✔ (V6 < 39.5)
|        | | ├✔ (V17 < 42.267137000000005)
|        | | | ├✔ (V17 < 37.5)
|        | | | | ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
|        | | | | └✘ 2 : (ninstances = 3, ncovered = 3, confidence = 1.0,
lift = 1.0)
|        | | | └✘ (V4 < 2.5)
|        | | |   ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0,
lift = 1.0)
|        | | |   └✘ (V2 < 13.5)
|        | | |     ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
|        | | |     └✘ 3 : (ninstances = 1, ncovered = 1, confidence = 1.
0, lift = 1.0)
|        | | └✘ (V19 < 78.914651999999999)
|        | |   ├✔ (V1 < 2010.9138755)
|        | |   | ├✔ 4 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
|        | |   | └✘ (V2 < 11.0)
|        | |   |   ├✔ 3 : (ninstances = 6, ncovered = 6, confidence = 1.
0, lift = 1.0)
|        | |   |   └✘ (V1 < 2015.0)
|        | |   |     ├✔ 2 : (ninstances = 3, ncovered = 3, confidence =
1.0, lift = 1.0)
|        | |   |     └✘ 3 : (ninstances = 1, ncovered = 1, confidence =
1.0, lift = 1.0)
|        | |   └✘ 4 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
|        | └✘ 1 : (ninstances = 10, ncovered = 10, confidence = 1.0, lif
t = 1.0)
|        └✘ (V1 < 2013.4190605)
|          ├✔ 2 : (ninstances = 10, ncovered = 10, confidence = 1.0, lif
t = 1.0)
|          └✘ (V1 < 2013.9190605)
|            ├✔ 4 : (ninstances = 1, ncovered = 1, confidence = 1.0, lif
t = 1.0)
|            └✘ (V13 < 89.0)
|              ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
|              └✘ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
├[9/10]┐ (V1 < 2016.5)
```

```
|       ├✔ (V19 < 73.92587900000001)
|       | ├✔ (V2 < 21.0)
|       | | ├✔ (V16 < 58.5)
|       | | | ├✔ 2 : (ninstances = 4, ncovered = 4, confidence = 1.0, lif
t = 1.0)
|       | | | └✘ (V4 < 2.0)
|       | | |   ├✔ 4 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
|       | | |   └✘ (V1 < 2011.3380240000001)
|       | | |     ├✔ (V21 < 42.6471995)
|       | | |     | ├✔ 2 : (ninstances = 2, ncovered = 2, confidence = 1.
0, lift = 1.0)
|       | | |     | └✘ 3 : (ninstances = 3, ncovered = 3, confidence = 1.
0, lift = 1.0)
|       | | |     └✘ 3 : (ninstances = 13, ncovered = 13, confidence = 1.
0, lift = 1.0)
|       | | └✘ (V15 < 72.0)
|       | |   ├✔ (V10 < 23.5)
|       | |   | ├✔ 2 : (ninstances = 1, ncovered = 1, confidence = 1.0, l
ift = 1.0)
|       | |   | └✘ 1 : (ninstances = 3, ncovered = 3, confidence = 1.0, l
ift = 1.0)
|       | |   └✘ 2 : (ninstances = 2, ncovered = 2, confidence = 1.0, lif
t = 1.0)
|       | └✘ (V2 < 1.5)
|       |   ├✔ 1 : (ninstances = 5, ncovered = 5, confidence = 1.0, lift
= 1.0)
|       |   └✘ (V2 < 6.5)
|       |     ├✔ (V6 < 50.266035)
|       |     | ├✔ 3 : (ninstances = 4, ncovered = 4, confidence = 1.0, l
ift = 1.0)
|       |     | └✘ 2 : (ninstances = 2, ncovered = 2, confidence = 1.0, l
ift = 1.0)
|       |     └✘ 2 : (ninstances = 13, ncovered = 13, confidence = 1.0, l
ift = 1.0)
|       └✘ 1 : (ninstances = 11, ncovered = 11, confidence = 1.0, lift =
1.0)
└[10/10]┐ (V4 < 2.5)
        ├✔ (V14 < 82.750595)
        | ├✔ (V2 < 8.0)
        | | ├✔ (V2 < 2.5)
        | | | ├✔ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, li
ft = 1.0)
        | | | └✘ 4 : (ninstances = 4, ncovered = 4, confidence = 1.0, li
ft = 1.0)
        | | └✘ 1 : (ninstances = 13, ncovered = 13, confidence = 1.0, li
ft = 1.0)
        | └✘ (V17 < 74.0949275)
        |   ├✔ 2 : (ninstances = 6, ncovered = 6, confidence = 1.0, lift
= 1.0)
        |   └✘ 1 : (ninstances = 1, ncovered = 1, confidence = 1.0, lift
= 1.0)
        └✘ (V1 < 2015.0)
          ├✔ (V17 < 39.0)
          | ├✔ (V15 < 67.5)
          | | ├✔ 3 : (ninstances = 2, ncovered = 2, confidence = 1.0, li
ft = 1.0)
          | | └✘ (V16 < 68.5407125)
          | |   ├✔ 4 : (ninstances = 3, ncovered = 3, confidence = 1.0,
lift = 1.0)
```

```
            │ │   └✘ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
            │ └✘ (V2 < 1.5)
            │   ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, li
ft = 1.0)
            │   └✘ (V10 < 24.706846)
            │     ├✔ (V18 < 48.0)
            │     │ ├✔ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0
, lift = 1.0)
            │     │ └✘ 2 : (ninstances = 13, ncovered = 13, confidence = 1
.0, lift = 1.0)
            │     └✘ (V16 < 79.4806935)
            │       ├✔ (V13 < 40.0)
            │       │ ├✔ 2 : (ninstances = 4, ncovered = 4, confidence = 1
.0, lift = 1.0)
            │       │ └✘ 3 : (ninstances = 2, ncovered = 2, confidence = 1
.0, lift = 1.0)
            │       └✘ 3 : (ninstances = 3, ncovered = 3, confidence = 1.0
, lift = 1.0)
            └✘ (V16 < 76.0)
              ├✔ (V16 < 70.5)
              │ ├✔ (V14 < 69.5)
              │ │ ├✔ 1 : (ninstances = 3, ncovered = 3, confidence = 1.0,
lift = 1.0)
              │ │ └✘ 3 : (ninstances = 1, ncovered = 1, confidence = 1.0,
lift = 1.0)
              │ └✘ 3 : (ninstances = 2, ncovered = 2, confidence = 1.0, li
ft = 1.0)
              └✘ 1 : (ninstances = 3, ncovered = 3, confidence = 1.0, lift
= 1.0)
```

In [33]:
```
listrules(🌲🌲🌲)
```

```
OutOfMemoryError()

Stacktrace:
  [1] GenericMemory
    @ ./boot.jl:516 [inlined]
  [2] new_as_memoryref
    @ ./boot.jl:535 [inlined]
  [3] Array
    @ ./boot.jl:588 [inlined]
  [4] Array
    @ ./boot.jl:594 [inlined]
  [5] similar
    @ ./abstractarray.jl:868 [inlined]
  [6] similar
    @ ./abstractarray.jl:867 [inlined]
  [7] _array_for
    @ ./array.jl:676 [inlined]
  [8] collect(itr::Base.Generator{Base.Iterators.Zip{Tuple{Base.Iterator
s.ProductIterator{NTuple{10, Vector{Rule{UInt32}}}}}}, IterTools.var"#3#
4"{SoleModels.var"#191#194"{Bool, DecisionEnsemble{UInt32, Branch{UInt32},
SoleModels.var"#81#91"{SoleModels.var"#81#82#92"{DecisionTreeExt.var"#1#
3"}}, Nothing}}}})
    @ Base ./array.jl:796
  [9] listrules(m::DecisionEnsemble{UInt32, Branch{UInt32}, SoleModels.va
r"#81#91"{SoleModels.var"#81#82#92"{DecisionTreeExt.var"#1#3"}}, Nothing};
compute_metrics::Bool, metrics_kwargs::@NamedTuple{}, use_shortforms::Boo
l, use_leftmostlinearform::Nothing, normalize::Bool, normalize_kwargs::@Na
medTuple{allow_atom_flipping::Bool, rotate_commutatives::Bool}, scalar_sim
plification::Bool, force_syntaxtree::Bool, min_coverage::Nothing, min_ncov
ered::Nothing, min_ninstances::Nothing, min_confidence::Nothing, min_lif
t::Nothing, metric_filter_callback::Nothing, kwargs::@Kwargs{})
    @ SoleModels ~/.julia/packages/SoleModels/bbNUZ/src/rule-extraction.j
l:223
 [10] listrules(m::DecisionEnsemble{UInt32, Branch{UInt32}, SoleModels.va
r"#81#91"{SoleModels.var"#81#82#92"{DecisionTreeExt.var"#1#3"}}, Nothing})
    @ SoleModels ~/.julia/packages/SoleModels/bbNUZ/src/rule-extraction.j
l:166
 [11] top-level scope
    @ In[33]:1
```

This probably caused an `OutOfMemoryError()` (on my machine, it sure did!)

To appreciate this last part, let's play with a smaller model.

In [34]:
```
forest = MLJDecisionTreeInterface.RandomForestClassifier(max_depth=3, n_t
```

Out[34]:
```
RandomForestClassifier(
  max_depth = 3,
  min_samples_leaf = 1,
  min_samples_split = 2,
  min_purity_increase = 0.0,
  n_subfeatures = -1,
  n_trees = 3,
  sampling_fraction = 0.7,
  feature_importance = :impurity,
  rng = TaskLocalRNG())
```

In [35]:
```
forestmach = machine(forest, X_train, y_train)
```

Out[35]:
```
untrained Machine; caches model-specific representations of data
  model: RandomForestClassifier(max_depth = 3, …)
  args:
    1:  Source @650 ⏎ Table{AbstractVector{Continuous}}
    2:  Source @281 ⏎ AbstractVector{OrderedFactor{4}}
```

In [36]:
```
MLJ.fit!(forestmach, verbosity=0)
```

Out[36]:
```
trained Machine; caches model-specific representations of data
  model: RandomForestClassifier(max_depth = 3, …)
  args:
    1:  Source @650 ⏎ Table{AbstractVector{Continuous}}
    2:  Source @281 ⏎ AbstractVector{OrderedFactor{4}}
```

In [37]:
```
🌱🌱 = fitted_params(forestmach).forest # \:seedling:
```

Out[37]:
```
Ensemble of Decision Trees
Trees:      3
Avg Leaves: 6.333333333333333
Avg Depth:  3.0
```

Let's evaluate its performance.

In [38]:
```
y_predict_probabilities = MLJ.predict(forestmach, X_test)
y_predict = mode.(y_predict_probabilities)
cm = confusion_matrix(y_predict, y_test)
```

Out[38]:

	Ground Truth			
Predicted	Medium R…	Low Risk	No Risk	High Risk
Medium R…	11	0	1	2
Low Risk	1	2	2	0
No Risk	0	2	3	0
High Risk	0	0	0	0

In [39]:
```
accuracy(cm)
```

Out[39]:
```
0.6666666666666666
```

Let's extract logical rules!

In [40]:
```
🌲🌲 = solemodel(🌱🌱)  # \:evergreen_tree:
```

```
Out[40]: ▣ Ensemble{UInt32} of 3 models of type Branch{UInt32}
         ├[1/3]┐ (V10 < 20.5)
         │      ├✔ 1 : (ninstances = 10, ncovered = 10, confidence = 1.0, lift =
         1.0)
         │      └✘ (V21 < 75.411651)
         │        ├✔ (V4 < 2.0)
         │        │ ├✔ 1 : (ninstances = 14, ncovered = 14, confidence = 0.57, lif
         t = 1.0)
         │        │ └✘ 3 : (ninstances = 20, ncovered = 20, confidence = 0.5, lift
         = 1.0)
         │        └✘ (V3 < 24.5)
         │          ├✔ 2 : (ninstances = 16, ncovered = 16, confidence = 0.62, lif
         t = 1.0)
         │          └✘ 3 : (ninstances = 5, ncovered = 5, confidence = 1.0, lift =
         1.0)
         ├[2/3]┐ (V13 < 78.0)
         │      ├✔ (V13 < 42.151293)
         │      │ ├✔ (V13 < 10.947654)
         │      │ │ ├✔ 3 : (ninstances = 16, ncovered = 16, confidence = 0.38, lif
         t = 1.0)
         │      │ │ └✘ 4 : (ninstances = 6, ncovered = 6, confidence = 0.67, lift
         = 1.0)
         │      │ └✘ (V18 < 57.7403355)
         │      │   ├✔ 3 : (ninstances = 3, ncovered = 3, confidence = 0.67, lift
         = 1.0)
         │      │   └✘ 3 : (ninstances = 6, ncovered = 6, confidence = 1.0, lift =
         1.0)
         │      └✘ (V1 < 2016.5)
         │        ├✔ (V14 < 72.5)
         │        │ ├✔ 3 : (ninstances = 10, ncovered = 10, confidence = 0.4, lift
         = 1.0)
         │        │ └✘ 2 : (ninstances = 17, ncovered = 17, confidence = 0.88, lif
         t = 1.0)
         │        └✘ 1 : (ninstances = 7, ncovered = 7, confidence = 1.0, lift =
         1.0)
         └[3/3]┐ (V17 < 39.690162)
                ├✔ (V4 < 2.0)
                │ ├✔ 1 : (ninstances = 4, ncovered = 4, confidence = 1.0, lift =
         1.0)
                │ └✘ (V1 < 2011.0)
                │   ├✔ 1 : (ninstances = 2, ncovered = 2, confidence = 1.0, lift =
         1.0)
                │   └✘ 3 : (ninstances = 8, ncovered = 8, confidence = 1.0, lift =
         1.0)
                └✘ (V4 < 1.5)
                  ├✔ (V21 < 75.83743150000001)
                  │ ├✔ 4 : (ninstances = 7, ncovered = 7, confidence = 0.71, lift
         = 1.0)
                  │ └✘ 2 : (ninstances = 10, ncovered = 10, confidence = 0.9, lift
         = 1.0)
                  └✘ (V1 < 2016.5)
                    ├✔ 2 : (ninstances = 24, ncovered = 24, confidence = 0.54, lif
         t = 1.0)
                    └✘ 1 : (ninstances = 10, ncovered = 10, confidence = 1.0, lift
         = 1.0)
```

```
In [41]: listrules(🌲🌲)
```

```
Out[41]: 5×7×7 Array{Rule{UInt32}, 3}:
[:, :, 1] =
 ▣ ((V10 < 20.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.94
7654)) ∧ ((V17 < 39.690162)) ∧ ((V4 < 2.0))  ↣  1
                                        …  ▣ ((V10 < 20.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 < 2.0))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4
< 2.0))  ↣  1
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 < 2.0))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4
< 2.0))  ↣  3
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 < 2.0))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4
< 2.0))  ↣  2
    ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 < 2.0))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4
< 2.0))  ↣  3
    ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 < 2.0))  ↣  1


[:, :, 2] =
 ▣ ((V10 < 20.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.94
7654)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 < 2011.0))  ↣  1
                                        …  ▣ ((V10 < 20.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 < 20
11.0))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4
≥ 2.0)) ∧ ((V1 < 2011.0))  ↣  1
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 < 20
11.0))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4
≥ 2.0)) ∧ ((V1 < 2011.0))  ↣  3
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 < 20
11.0))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4
≥ 2.0)) ∧ ((V1 < 2011.0))  ↣  2
    ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 < 20
11.0))  ↣  1
```

▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 < 2011.0)) ↣ 3
▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 < 2011.0)) ↣ 1

[:, :, 3] =
▣ ((V10 < 20.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 3
… ▣ ((V10 < 20.5)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 1

▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 3
▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 1

▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 3
▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 3

▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 3
▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 2

▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 3
▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 < 39.690162)) ∧ ((V4 ≥ 2.0)) ∧ ((V1 ≥ 2011.0)) ↣ 3

[:, :, 4] =
▣ ((V10 < 20.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 < 75.83743150000001)) ↣ 4
… ▣ ((V10 < 20.5)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 < 75.83743150000001)) ↣ 1

▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 < 75.83743150000001)) ↣ 4
▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 ≥ 78.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 < 75.83743150000001)) ↣ 1

▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 < 78.0))

```
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
< 1.5)) ∧ ((V21 < 75.83743150000001))  ↣  3
      ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 < 7
5.83743150000001))  ↣  4

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
< 1.5)) ∧ ((V21 < 75.83743150000001))  ↣  4
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 < 7
5.83743150000001))  ↣  4

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
< 1.5)) ∧ ((V21 < 75.83743150000001))  ↣  3
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 < 7
5.83743150000001))  ↣  4


[:, :, 5] =
 ▣ ((V10 < 20.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.94
7654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 ≥ 75.8374315000000
1))  ↣  2
                                        …  ▣ ((V10 < 20.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 ≥ 7
5.83743150000001))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
< 1.5)) ∧ ((V21 ≥ 75.83743150000001))  ↣  2
      ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 ≥ 7
5.83743150000001))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
< 1.5)) ∧ ((V21 ≥ 75.83743150000001))  ↣  3
      ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 ≥ 7
5.83743150000001))  ↣  2

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
< 1.5)) ∧ ((V21 ≥ 75.83743150000001))  ↣  2
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 ≥ 7
5.83743150000001))  ↣  2

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
< 1.5)) ∧ ((V21 ≥ 75.83743150000001))  ↣  3
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 < 1.5)) ∧ ((V21 ≥ 7
5.83743150000001))  ↣  2


[:, :, 6] =
 ▣ ((V10 < 20.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.94
```

```
7654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 < 2016.5))  ↣  2
                                        …  ▣ ((V10 < 20.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 < 20
16.5))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 < 2016.5))  ↣  2
      ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 < 20
16.5))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 < 2016.5))  ↣  3
      ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 < 20
16.5))  ↣  2

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 < 2016.5))  ↣  2
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 < 20
16.5))  ↣  2

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 < 2016.5))  ↣  3
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 < 20
16.5))  ↣  2


[:, :, 7] =
 ▣ ((V10 < 20.5)) ∧ ((V13 < 78.0)) ∧ ((V13 < 42.151293)) ∧ ((V13 < 10.94
7654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 ≥ 2016.5))  ↣  1
                                        …  ▣ ((V10 < 20.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 ≥ 20
16.5))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 ≥ 2016.5))  ↣  1
      ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 < 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 ≥ 20
16.5))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 ≥ 2016.5))  ↣  3
      ▣ ((V10 ≥ 20.5)) ∧ ((V21 < 75.411651)) ∧ ((V4 ≥ 2.0)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 ≥ 20
16.5))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 ≥ 2016.5))  ↣  2
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 < 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 ≥ 20
```

```
16.5))  ↣  1

 ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 < 78.0))
∧ ((V13 < 42.151293)) ∧ ((V13 < 10.947654)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4
≥ 1.5)) ∧ ((V1 ≥ 2016.5))  ↣  3
     ▣ ((V10 ≥ 20.5)) ∧ ((V21 ≥ 75.411651)) ∧ ((V3 ≥ 24.5)) ∧ ((V13 ≥ 7
8.0)) ∧ ((V1 ≥ 2016.5)) ∧ ((V17 ≥ 39.690162)) ∧ ((V4 ≥ 1.5)) ∧ ((V1 ≥ 20
16.5))  ↣  1
```

Let's summarize our model joining rules associated with the same class!

In [42]:
```
metricstable(joinrules(🌲🌲; min_ncovered = 1, normalize = true))
```

Antecedent	Consequent	ninstances	ncovered	coverage	confidence	lift	natoms

```
((V10 < 20.5) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V1
0 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 < 10.947654) ∧ (V17 < 3
9.690162) ∧ (V4 < 2.0)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧
(V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4
< 2.0) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨
((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 3
9.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧
(V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 <
2.0)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V
17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 <
2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧
(V4 < 2.0)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5)
∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V
4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162)
∧ (V4 < 2.0)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 7
2.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651)
```

∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690
162) ∧ (V4 < 2.0)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 <
39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0)
∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥
20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V
17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 <
24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨
((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 201
6.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 < 20.5) ∧ (V13 < 10.947654)
∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 <
75.411651) ∧ (V4 < 2.0) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 ≥
2.0) ∧ (V1 < 2011.0)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V
17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 7
5.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162)
∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧
(V18 < 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V
10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧
(V18 < 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V
10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 < 39.690
162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧
(V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 < 39.690
162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 <
2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨
((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 201
6.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨
((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.69
0162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧
(V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.69016
2) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 20
16.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧
(V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 3
9.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.41165
1) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥
2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧
(V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 201
1.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧
(V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 <
20.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V
1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 7
8.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨
((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 <
1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V
4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧
(V21 < 75.83743150000001)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5)
∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 2
0.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V1
7 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 < 20.5) ∧
(V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 201
6.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V
1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 < 2
0.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5))
∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 < 10.947654) ∧ (V1
7 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [1
0.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨
((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [10.947654,42.1512
93)) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 < 20.5) ∧
(V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧
(V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥

```
57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 2
0.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥
57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 < 2
0.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧
(V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 <
2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧
(V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5)
∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥
20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V
14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 < 2
0.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1
≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0)
∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10
≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧
(V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 7
5.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162)
∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥
24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V
1 ≥ 2016.5)) │          1 │       3835 │      370 │ 0.0964798 │   0.924324
│   2.861 │    405 │
│ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 < 10.947654) ∧ (V1
7 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 2
4.5) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5)
∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.74
03355) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.41165
1) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 <
39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0)
∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4
< 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151
293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10
≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧
(V14 < 72.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 7
5.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V
17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥
2.0) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0))
∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 < 10.947654) ∧ (V
17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 7
5.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧
(V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 7
5.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧
(V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 7
5.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧
(V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 7
5.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧
(V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 7
5.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V1
7 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.4
11651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17
< 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 < 20.5) ∧ (V13 < 10.947
654) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧
(V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥
2.0) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0))
∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 < 10.947654) ∧ (V
17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 7
5.411651) ∧ (V3 ≥ 24.5) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 ≥
2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧
(V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 201
1.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [10.94765
4,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 <
```

20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162)
∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 <
2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥
2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 2
4.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 2
4.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧
(V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V
10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧
(V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V
10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧
(V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V
10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧
(V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V
10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧
(V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V
10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.69016
2) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4
< 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥
2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 2
4.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 2
4.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥
2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 2
4.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧
(V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥
2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1
≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 7
8.0) ∧ (V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨
((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 < 10.947654) ∧ (V17
≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧
(V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧
(V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.41165
1) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 3
9.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21
≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355)
∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 2
0.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥
57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001))
∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,7
8.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83
743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 7
8.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V2
1 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5)
∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 <
1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V
4 ≥ 2.0) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 7
5.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V1
3 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.8374315000000
1)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,7
8.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83
743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈
[42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧

(V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) | 3 | 4810 | 664 | 0.138046 | 0.831325 | 3.1786 | 576 |
|
((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 < 10.947654) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411

651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 <
39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.41165
1) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 3
9.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.41165
1) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 3
9.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.41165
1) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 3
9.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧
(V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 201
1.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V
1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 201
1.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧
(V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 201
1.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧
(V1 ≥ 2016.5) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥
20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧
(V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001))
∨ ((V10 < 20.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V2
1 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧
(V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000
001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 < 10.94765
4) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥
20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧
(V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 < 2
0.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧
(V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.41165
1) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 3
9.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21
≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355)
∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 < 2
0.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧
(V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.41165
1) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 3
9.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21
≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355)
∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 < 2
0.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧
(V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.41165
1) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 3
9.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21
≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧
(V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 < 2
0.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧
(V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.41165
1) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 3
9.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21
< 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧
(V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10 ≥ 2
0.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V
14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨
((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 201
6.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.837431500
00001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧
(V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.8374315000000
1)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1
≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨
((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 201
6.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150000001)) ∨ ((V10
< 20.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 201
6.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 < 10.947654)

∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥
75.411651) ∧ (V3 < 24.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧
(V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 201
6.5)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V
17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 7
5.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧
(V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 7
5.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧
(V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 < 20.5) ∧ (V13 ∈
[42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧
(V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ∈
[42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧
(V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈
[42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧
(V1 < 2016.5)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 7
2.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧
(V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 7
2.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧
(V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 7
2.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 < 20.5) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧
(V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 201
6.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧
(V1 ≥ 2016.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥
20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧
(V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 7
5.411651) ∧ (V3 < 24.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧
(V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧
(V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧
(V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥
1.5) ∧ (V1 ≥ 2016.5)) │ 2 │ 4420 │ 865 │ 0.195701 │
0.706358 │ 2.08697 │ 536 │
│
((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [10.947654,42.1512
93)) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651)
∧ (V3 < 24.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 <
2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [10.94765
4,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 < 2.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 7
5.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162)
∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 <
24.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧

(V1 < 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈
[10.947654,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 < 2011.0)) ∨
((V10 < 20.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 ≥
2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧
(V13 ∈ [10.947654,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 201
1.0)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [10.94765
4,42.151293)) ∧ (V17 < 39.690162) ∧ (V4 ≥ 2.0) ∧ (V1 ≥ 2011.0)) ∨ ((V10 <
20.5) ∧ (V13 < 10.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.837
43150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 < 1
0.947654) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨
((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 < 10.947654) ∧ (V17
≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 < 20.5) ∧
(V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 7
5.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13
∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.8374
3150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [1
0.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150
000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [10.94
7654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.837431500000
01)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [10.94765
4,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.8374315000000
1)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403355) ∧ (V17
≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧
(V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 < 57.7403
355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10
≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V1
8 < 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.8374315000000
1)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403355) ∧ (V17
≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧
(V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V18 ≥ 57.7403
355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10
≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ∈ [42.151293,78.0)) ∧ (V1
8 ≥ 57.7403355) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.8374315000000
1)) ∨ ((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5) ∧ (V17 ≥
39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V2
1 < 75.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 < 72.5)
∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 2
0.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V
14 < 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨
((V10 < 20.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V17 ≥ 39.69
0162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 7
5.411651) ∧ (V4 < 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 72.5) ∧ (V1
7 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧
(V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧ (V14 ≥ 7
2.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10
≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 < 2016.5) ∧
(V14 ≥ 72.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001))
∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 20
16.5) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V1
0 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 < 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5)
∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 ≥ 2
0.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ≥ 78.0) ∧ (V1 ≥ 2016.5) ∧ (V
17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 < 75.83743150000001)) ∨ ((V10 < 20.5)
∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥
75.83743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 < 2.0) ∧ (V1
3 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83
743150000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈
[10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.837431
50000001)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [1
0.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 < 1.5) ∧ (V21 ≥ 75.83743150

000001)) ∨ ((V10 < 20.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.69016
2) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4
< 2.0) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧
(V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈
[10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨
((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5) ∧ (V13 ∈ [10.947654,42.151
293)) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 < 2016.5)) ∨ ((V10 ≥ 20.5) ∧
(V21 < 75.411651) ∧ (V4 ≥ 2.0) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 3
9.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.41165
1) ∧ (V3 < 24.5) ∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4
≥ 1.5) ∧ (V1 ≥ 2016.5)) ∨ ((V10 ≥ 20.5) ∧ (V21 ≥ 75.411651) ∧ (V3 ≥ 24.5)
∧ (V13 ∈ [10.947654,42.151293)) ∧ (V17 ≥ 39.690162) ∧ (V4 ≥ 1.5) ∧ (V1 ≥ 2
016.5)) │ 4 │ 2860 │ 376 │ 0.131469 │ 0.744681 │ 9.
68085 │ 310 │
└