Shock & Awe aids

“How to be a Data Scientist Impostor?” presentation at OMLDS 2019-10-5

Anton Antonov
October 2019

Image restyling

Finding textual answers

Here is a generic example.

In[∘]:=
```
FindTextualAnswer[
 "Paris is the capital and most populous city of France, with a 2015 population of 2,229,621.",
 "How many people live in Paris?"]
```

Out[∘]= 2,229,621

Answers from Wikipedia articles

Here is a more complicated example with Wikipedia data.

Orlando

In[∘]:=
```
question = "Who is the current mayor of Orlando?";
```

In[∘]:=
```
articles = WikipediaData /@ WikipediaSearch["Content" → question, "MaxItems" → 6];
```

In[∘]:=
```
GridTableForm[FindTextualAnswer[articles, question, 3, {"Probability", "HighlightedSentence"}]]
```

Out[∘]=

#	1	2
1	0.921165	The current mayor is Buddy Dyer, who was first elected in a special election in February 2003.
2	0.823615	In his years as mayor of Orlando, Buddy Dyer claims progress in realizing his vision for Orlando as a "world-class city."
3	0.816201	The first mayor, William Jackson Brack, took office in 1875.

In[∘]:=
```
GridTableForm[FindTextualAnswer[articles, "What is Orlando's population", 3, {"Probability", "HighlightedSentence"}]]
```

Out[∘]=

#	1	2
1	0.92858	Located in Central Florida, it is the center of the Orlando metropolitan area, which had a population of 2,509,831, according to U.S. Census Bureau figures released in July 2017.
2	0.750153	The area encompasses four counties (Orange, Osceola, Seminole and Lake), and is the 26th-largest metro area in the United States with a 2010 Census-estimated population of 2,134,411.In 2000, the population of Orlando's urban area was 1,157,431, making it the third-largest in Florida and the 35th-largest in the United States.
3	0.729993	As of 2009, the estimated urban area population of Orlando is 1,377,342.

WDW

In[∘]:=
```
question = "How big is World Disney World?";
```

In[∘]:=
```
articles = WikipediaData /@ WikipediaSearch["Content" → question, "MaxItems" → 6];
```

In[∘]:=
```
GridTableForm[FindTextualAnswer[articles, question, 3, {"Probability", "HighlightedSentence"}]]
```

Out[∘]=

#	1	2
1	0.772294	The property, which covers nearly 25,000 acres (39 sq mi;
2	0.732221	To avoid a burst of land speculation, Walt Disney World Company used various dummy corporations to acquire 30,500 acres (48 sq mi;
3	0.584723	Walt Disney World requires an estimated 1 billion kilowatt-hours (3.6 billion megajoules) of electricity annually, costing the company nearly $100 million in annual energy consumption.

In[•]:=
```
GridTableForm[
 FindTextualAnswer[articles, "What is the average number of WDW guests?", 3, {"Probability", "HighlightedSentence"}]]
```

Out[•]=

#	1	2
1	0.823858	Another one named Bonzai was responsible for the creation of the city's 250,000 trees, while a new rendering system called Hyperion offered new illumination possibilities, like light shining through a translucent object (e.g. Baymax's vinyl covering).
2	0.823023	The review aggregation website Rotten Tomatoes reports that 89% of critics gave the film a positive review based on 218 reviews, with an average score of 7.34/10.
3	0.793399	Today, Walt Disney World is the most visited vacation resort in the world, with average annual attendance of more than 52 million.

ClCon

In[•]:=
```
RecordsSummary[dfTitanic]
```

Out[•]=

1 id

1	1
10	1
100	1
1000	1
1001	1
1002	1
(Other)	1303

2 passengerClass

3rd	709
1st	323
2nd	277

3 passengerAge

Min	-1
1st Qu	10
Median	20
Mean	23.55
3rd Qu	40
Max	80

4 passengerSex

male	843
female	466

5 passengerSurvival

died	809
survived	500

In[•]:=
```
ClConUnit[dfTitanic] ⟹
  ClConSplitData[0.7] ⟹
  ClConEchoDataSummary ⟹
  ClConMakeClassifier["RandomForest"] ⟹
  ClConClassifierMeasurements[] ⟹
  ClConEchoValue ⟹
  ClConROCPlot[{"FPR", "TPR"}, ImageSize → Medium] ⟹
  ClConAccuracyByVariableShuffling ⟹
  ClConEchoValue;
```

» summaries: {trainingData →

1 id

10	1
100	1
1000	1
1001	1
1002	1
1003	1
(Other)	910

2 passengerClass

3rd	501
1st	224
2nd	191

3 passengerAge

Min	-1
1st Qu	10
Median	20
Mean	23.7325
3rd Qu	40
Max	70

4 passengerSex

male	591
female	325

5 passengerSurvival

died	566
survived	350

},

testData →

1 id

1	1
1007	1
1013	1
1014	1
1018	1
1020	1
(Other)	387

2 passengerClass

3rd	208
1st	99
2nd	86

3 passengerAge

Min	-1
1st Qu	0
Median	20
Mean	23.1247
3rd Qu	40
Max	80

4 passengerSex

male	252
female	141

5 passengerSurvival

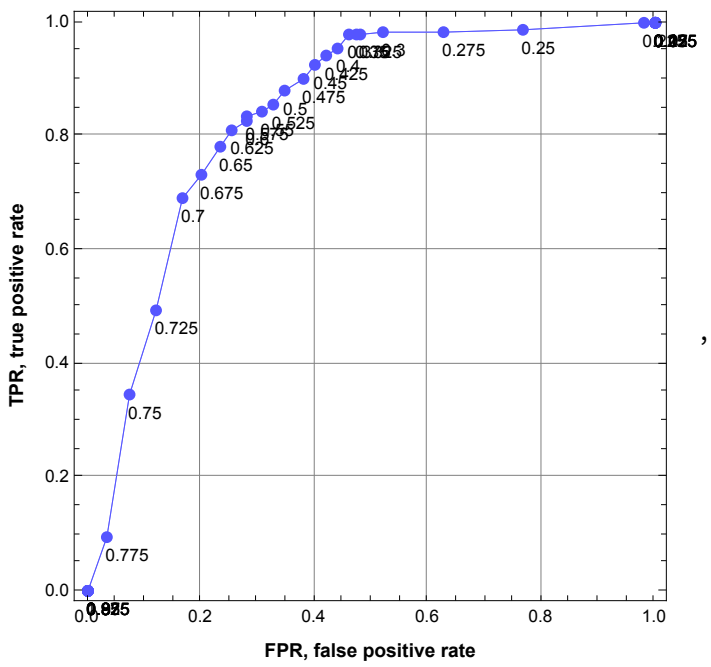
died	243
survived	150

}}

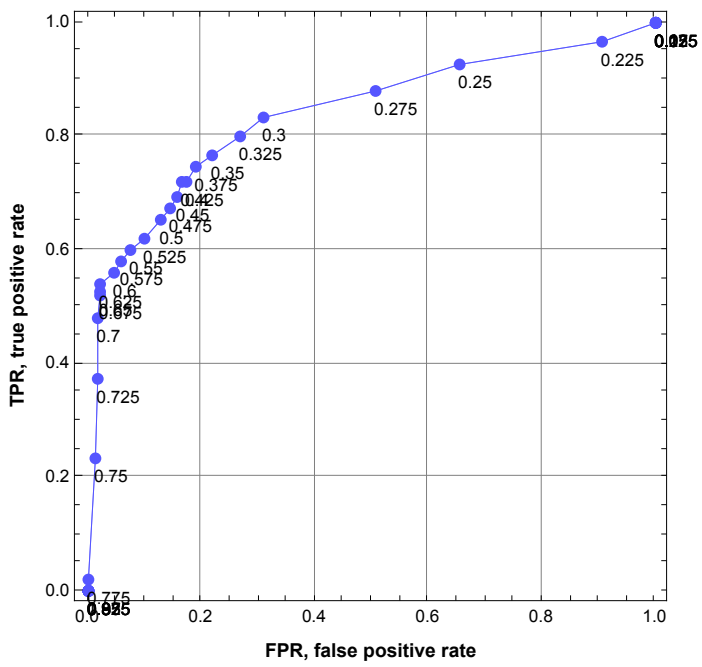
» value: ⟨|Accuracy → 0.788804, Precision → ⟨|died → 0.80303, survived → 0.75969|⟩, Recall → ⟨|died → 0.872428, survived → 0.653333|⟩|⟩

ClassifierInformation: ClassifierInformation is obsolete. It has been superseeded by Information since version 12.

» ROC plot(s): ⟨|died →



, survived →



|⟩

» value: ⟨|None → 0.788804, id → 0.801527, passengerClass → 0.737913, passengerAge → 0.804071, passengerSex → 0.549618|⟩

ML code generation : SMRMon

Load packages

In[∘]:=
```
Get["/Volumes/Macintosh HD/Users/antonov/ConversationalAgents/Packages/WL/ExternalParsersHookup.m"]
Get["/Volumes/Macintosh HD/Users/antonov/MathematicaForPrediction/MonadicProgramming/MonadicSparseMatrixRecommender.m"]
Get["/Volumes/Macintosh HD/Users/antonov/MathematicaForPrediction/MonadicProgramming/MonadicAnomaliesFinder.m"]
```

Load data

In[∘]:=
```
dfTitanic = Import["https://github.com/antononcube/MathematicaVsR/raw/master/Data/MathematicaVsR-Data-Titanic.csv"];
dfTitanic = Dataset[Rest[dfTitanic]][All, AssociationThread[First[dfTitanic] → #] &];

dfMushroom =
  Import["https://raw.githubusercontent.com/antononcube/MathematicaVsR/master/Data/MathematicaVsR-Data-Mushroom.csv"];
dfMushroom = Dataset[Rest[dfMushroom]][All, AssociationThread[First[dfMushroom] → #] &];
```

In[∘]:=
```
dfTitanic = dfTitanic[All, Prepend[#, "id" → ToString[#id]] &];
dfMushroom = dfMushroom[All, Prepend[#, "id" → ToString[#id]] &];
```

Examples

In[∘]:=
```
smrTitanic = SMRMonUnit[] ⟹ SMRMonCreate[dfTitanic, "id"] ⟹ SMRMonRecommend[{"1", "10"}, 12] ⟹ SMRMonEchoValue;
```

» value:

$$\langle\left|62 \to 1,\ 15 \to 1,\ 82 \to \frac{1751}{2001},\ 286 \to \frac{1751}{2001},\ 136 \to \frac{1751}{2001},\ 728 \to \frac{1167}{1334},\ 595 \to \frac{1167}{1334},\ 507 \to \frac{1167}{1334},\ 1236 \to \frac{1167}{1334},\ 84 \to \frac{1501}{2001},\ 80 \to \frac{1501}{2001},\ 79 \to \frac{1501}{2001}\right|\rangle$$

Recommendations by history

In[∘]:=
```
smrTitanic =
  ToSMRMonWLCommand["
create from dfTitanic;
compute 12 recommendations for the history 1, 10;
echo pipeline value", True];
```

» SMRMonCreate: Heuristically picking the ID column to be "id".

» value:

$$\langle\left|62 \to 1,\ 15 \to 1,\ 82 \to \frac{1751}{2001},\ 286 \to \frac{1751}{2001},\ 136 \to \frac{1751}{2001},\ 728 \to \frac{1167}{1334},\ 595 \to \frac{1167}{1334},\ 507 \to \frac{1167}{1334},\ 1236 \to \frac{1167}{1334},\ 84 \to \frac{1501}{2001},\ 80 \to \frac{1501}{2001},\ 79 \to \frac{1501}{2001}\right|\rangle$$

In[∘]:=
```
ToSMRMonWLCommand["
create from dfTitanic;
recommnd for the histry 1, 10;
echo pipeline value", False]
```

Out[∘]=
```
Possible misspelling of 'recommend' as 'recommnd'.
Possible misspelling of 'recommend' as 'recommnd'.
Possible misspelling of 'history' as 'histry'.
SMRMonUnit[] ⟹ SMRMonCreate[dfTitanic] ⟹
SMRMonRecommend[{"1", "10"}] ⟹
SMRMonEchoValue[]
```

In[∘]:=
```
ToSMRMonWLCommand["
use the recommender smrTitanic;
classify to passengerSurvival the profile male, 3rd;
echo pipeline value", True];
```

» value: $\langle\left|\text{died} \to 1,\ \text{survived} \to \frac{5}{17}\right|\rangle$

In[∘]:=
```
ToSMRMonWLCommand["
use recommender object smrTitanic;
compute 12 recommendations for the history 1, 14;
echo pipeline value;
join across with dfTitanic;
echo pipeline value
", True];
```

» value: $\langle\left|62 \to 1, 15 \to \frac{2402}{2403}, 82 \to \frac{2101}{2403}, 286 \to \frac{2101}{2403}, 136 \to \frac{2101}{2403}, 10 \to \frac{2101}{2403}, 728 \to \frac{700}{801}, 595 \to \frac{700}{801}, 507 \to \frac{700}{801}, 1236 \to \frac{700}{801}, 84 \to \frac{601}{801}, 80 \to \frac{601}{801}\right|\rangle$

» SMRMonJoinAcross: Heuristically picking the joining column to be "id".

» value:

Score	Item	passengerClass	passengerAge	passengerSex	passengerSurvival
1	62	1st	80	female	survived
0.999584	15	1st	80	male	survived
0.874324	10	1st	70	male	died
0.874324	136	1st	70	male	died
0.874324	286	1st	70	male	died
0.874324	82	1st	70	male	died
0.873908	1236	3rd	70	male	died
0.873908	507	2nd	70	male	died
0.873908	595	2nd	70	male	died
0.873908	728	3rd	70	male	died
0.750312	80	1st	60	female	survived
0.750312	84	1st	60	female	survived

Recommendations by profile

In[•]:=
```
ToSMRMonWLCommand["
use recommender object smrTitanic;
compute 12 recommendations for the profile male, survived;
echo pipeline value;
join across with dfTitanic;
echo pipeline value
", True];
```

» value: ‹|992 → 1., 986 → 1., 982 → 1., 979 → 1., 971 → 1., 954 → 1., 950 → 1., 95 → 1., 946 → 1., 942 → 1., 94 → 1., 936 → 1.|›

» SMRMonJoinAcross: Heuristically picking the joining column to be "id".

» value:

Score	Item	passengerClass	passengerAge	passengerSex	passengerSurvival
1.0	936	3rd	30	male	survived
1.0	94	1st	50	male	survived
1.0	942	3rd	20	male	survived
1.0	946	3rd	-1	male	survived
1.0	95	1st	0	male	survived
1.0	950	3rd	30	male	survived
1.0	954	3rd	20	male	survived
1.0	971	3rd	20	male	survived
1.0	979	3rd	30	male	survived
1.0	982	3rd	30	male	survived
1.0	986	3rd	20	male	survived
1.0	992	3rd	-1	male	survived

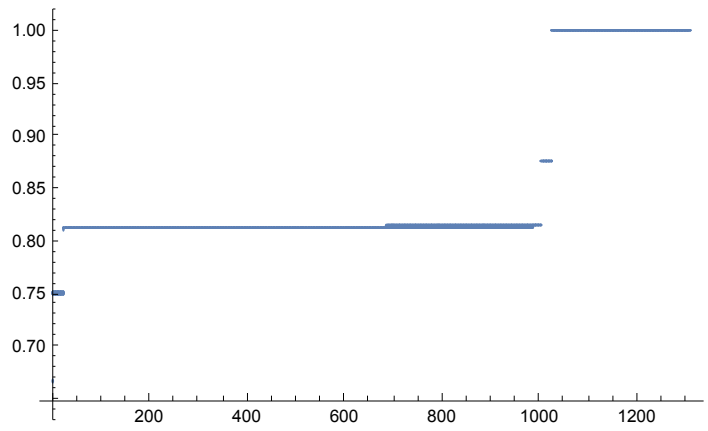
In[•]:=
```
ToSMRMonWLCommand["
use recommender object smrTitanic;
compute 12 recommendations for the profile male, survived;
echo pipeline value;
join across with dfTitanic;
echo pipeline value
", False]
```

Out[•]=
```
smrTitanic ⟹
SMRMonRecommendByProfile[{"male", "survived"},  12] ⟹
SMRMonEchoValue[] ⟹
SMRMonJoinAcross[dfTitanic] ⟹
SMRMonEchoValue[]
```

Finding anomalies

In[•]:=
```
smrRes =
  ToSMRMonWLCommand["
use recommender object smrTitanic;
find anomalies using 20 nearest neighbors, the aggregation function Median and the property Distances;
echo pipeline value
", True];
```

In[•]:=
```
smrRes ⟹ SMRMonEchoFunctionValue[ListPlot[Sort@N[Values[#]], PlotRange → All] &];
```



ML code generation : QRMon

Load packages

In[•]:=
```
Get["/Volumes/Macintosh HD/Users/antonov/ConversationalAgents/Packages/WL/ExternalParsersHookup.m"]
Get["/Volumes/Macintosh HD/Users/antonov/MathematicaForPrediction/MonadicProgramming/MonadicStructuralBreaksFinder.m"]
Get["/Volumes/Macintosh HD/Users/antonov/MathematicaForPrediction/MonadicProgramming/MonadicAnomaliesFinder.m"]
```

Distribution data

Temperature data

Financial data

Experiments

In[•]:=
```
ToQRMonWLCommand[
 "create from tsData; rescale both axes; echo data summary; compute quantile regression with 12 knots", False]
```

Out[•]=
```
QRMonUnit[tsData] ⟹
QRMonRescale["Axes"->{True, True}] ⟹
QRMonEchoDataSummary[] ⟹
QRMonQuantileRegression["Knots" -> 12]
```

In[•]:=
```
AbsoluteTiming[
 ToQRMonWLCommand["
create from tsData;
delete missing;
rescale regressor axis;
rescale value axis;
summarize data;
do quantile regression with 12 knots;
show plot;
find outliers;
", False]
]
```

Out[•]=
```
{0.211836, QRMonUnit[tsData] ⟹
QRMonDeleteMissing[] ⟹
QRMonRescale["Axes"->{True, False}] ⟹
QRMonRescale["Axes"->{False, True}] ⟹
QRMonEchoDataSummary[] ⟹
QRMonQuantileRegression["Knots" -> 12] ⟹
QRMonPlot[] ⟹
QRMonOutliersPlot[]
}
```

In[•]:=
```
ToQRMonWLCommand["
create from data;
rescale both axes;
echo summary;
compute quantile regression with 12 knots;
show plot
"];
```

» GetData: Cannot find data.

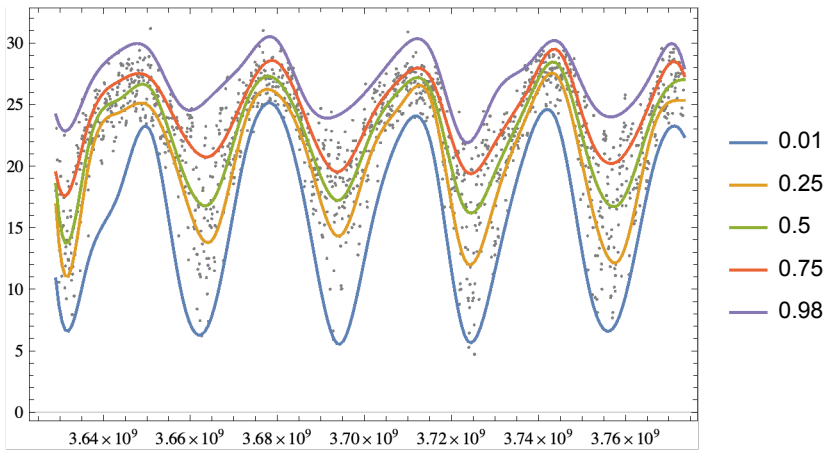
» QRMonBind: Failure when applying: QRMonRescale[Axes -> {True, True}]

In[•]:=
```
qrmon2 = ToQRMonWLCommand["
create from tsData;
delete missing;
echo data summary;
compute quantile regression with 20 knots and probabilities 0.01, 0.25, 0.5, 0.75, 0.98;
show date list plot;
compute and display outliers
", True];
```

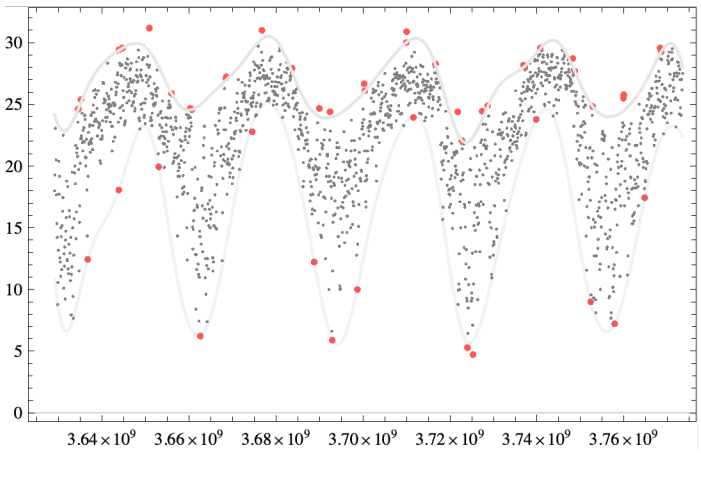
» Data summary:

1 column 1		2 column 2	
Min	3.62906×10^9	Min	4.72
1st Qu	3.66522×10^9	1st Qu	19.44
Mean	3.70132×10^9	Mean	22.2235
Median	3.70133×10^9	Median	23.28
3rd Qu	3.73745×10^9	3rd Qu	26.06
Max	3.77352×10^9	Max	31.17

» Plot:



» Outliers plot:

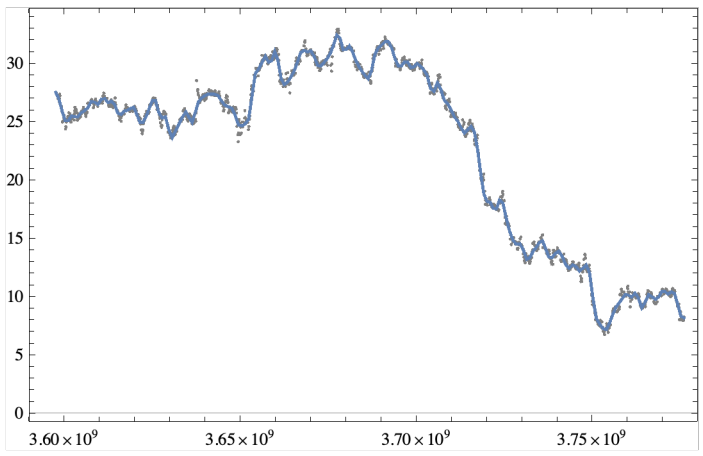


In[•]:=
```
qrmon2 = ToQRMonWLCommand["
create from finData;
delete missing;
echo data summary;
compute quantile regression with 120 knots and probabilities 0.5;
show date list plot;
show absolute errors plot;
find anomalies by the threshold 1;
echo pipeline value
", True];
```

» Data summary:

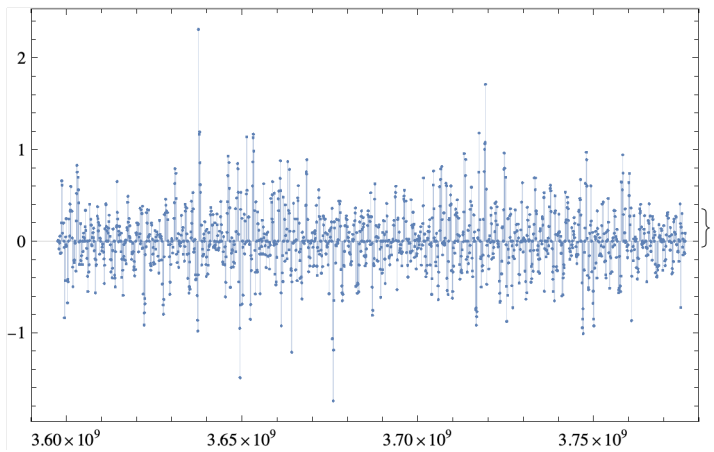
1 column 1		2 column 2	
Min	3.59761×10^9	Min	6.71
1st Qu	3.64226×10^9	1st Qu	14.38
Median	3.6866×10^9	Mean	22.6887
Mean	3.68683×10^9	Median	25.74
3rd Qu	3.73164×10^9	3rd Qu	28.9225
Max	3.77611×10^9	Max	32.93

» Plot:



0.5

» Error plots: {0.5 →



}

» value: $\{\{3.63761\times10^9, 28.51\}, \{3.63787\times10^9, 27.63\}, \{3.63796\times10^9, 27.73\}, \{3.64945\times10^9, 23.27\}, \{3.65135\times10^9, 25.93\}, \{3.65316\times10^9, 27.77\}, \{3.65325\times10^9, 28.03\}, \{3.66414\times10^9, 27.45\}, \{3.67572\times10^9, 29.82\}, \{3.67597\times10^9, 29.32\}, \{3.67606\times10^9, 29.94\}, \{3.71745\times10^9, 23.83\}, \{3.719\times10^9, 20.21\}, \{3.71909\times10^9, 20.12\}, \{3.71917\times10^9, 19.99\}, \{3.71926\times10^9, 20.49\}, \{3.74708\times10^9, 11.29\}\}$

Unit test running

ClCon

In[∘]:=
```
trObj = TestReport[
  "/Volumes/Macintosh HD/Users/antonov/MathematicaForPrediction/UnitTests/MonadicContextualClassification-Unit-Tests.wlt"]
```

Out[∘]= TestReportObject[Title: Test Report: MonadicContextualClassification−Unit−Tests.wlt
Success rate: 93% Tests run: 28]

In[∘]:=
```
trObj["TestsFailed"]
```

Out[∘]= ⟨|TestsFailedWrongResults → ⟨|19 → TestResultObject[Outcome: Failure
Test ID: AssignVariableNames−3]|⟩,

TestsFailedWithMessages → ⟨|24 → TestResultObject[Outcome: MessagesFailure
Test ID: Partial−data−summaries−1]|⟩, TestsFailedWithErrors → ‹| |›|⟩