### Paul Harten<sup>1</sup>, Henry Helgen<sup>2</sup>, Wilson Melendez<sup>2</sup>

<sup>1</sup>US EPA/ORD, Center for Computational Toxicology and Exposure; <sup>2</sup>General Dynamics Information Technology, Contractor to US EPA

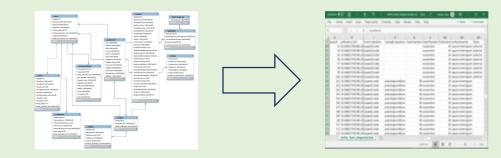
#### **OBJECTIVES**

#### APPROACH

#235

- Datamine EPA's database NaKnowBase that contains details about thousands of experiments involving nanomaterials conducted by ORD scientists.
- Predict the adverse effects of nanomaterials from models built from a rich number of examples via regression analysis.

Flatten the relational database NaKnowBase, pivoting on tables to place multiple entries for each experiment into one unique row, with all experiments placed in a spreadsheet format.



#### **MAIN RESULTS**

- EPA's relational database NaKnowBase is flattened into a spreadsheet format. Other organizations can copy and modify the nanomaterial data for their own analysis.
- Initial steps for EPA's Chemical Safety for Sustainability (CSS), Emerging Materials and Technology (EMT) CSS 3.2.3 Predictive model - nanoQSAR

#### IMPACT

- Approaches to develop nanoQSAR models from relational databases will facilitate the prediction of adverse activities of novel nanomaterials.
- EPA Office of Research and Development can use nanoQSAR models for additional investigative methods.
- For more information, contact: Paul Harten, harten.paul@epa.gov

This work does not reflect EPA policy.

# **Overview**

- Most regression analysis algorithms have training and testing samples given in spreadsheet format, a sample for each row.
- However, sometimes the parameters, variables, and results of experiments are held in relational databases to reduce the space needed to itemize corresponding features.
- Because of the structure of relational databases, datamining examples of detailed experiments into a spreadsheet format can be particularly difficult.
- This method translates your relational database into a spreadsheet format CSV file ready for regression analysis.

### **APPROACH**

Flatten the relational database NaKnowBase, pivoting on tables to place multiple entries for each experiment into one unique row, with all experiments placed in a spreadsheet format.

			Courses and a		
Contract of the second se	6 · · · · · · · · · · · · · · · · · · ·		Annual of	Charlestown 1	
Country of			Tradicator (cit matteries)	Contraction of the local sector	
And the second s			Index of the local division of the local div		
The Avenue of th	personal second second second		· Technical and a state of the	Texture of the local division of the local d	
Transportation of Address (1993)			Cample and Without		
Property in the second second			Participation (Contraction)	Canada	
Tranks patient, pt 100000			Contraction (Second ed.)	and the second s	
Property Statements and	Contraction of the second seco		1000 100000	reasoning in the second states	
The second		and the second se	Pages Included	of Astronomy States March 18.	
the second second second second second		and a second second	National address	And the second se	
			Calcular Description	A CONTRACTOR OF A CONTRACTOR	
A CONTRACT OF		Car and a second second	- Reference And		100
		Second Second	International Advancements	and the second se	
			- Andrewski and		
			"had a second second	Transfer 1	100
and the second s		A COLORED TO A	Contraction and Contract	the second second	
100000000000000000000000000000000000000			The second second second second	Contract of Contract of	
		A DECISION OF A	"hadronic televised	Contractional admittation	
	and the second se		International Action (1997)	a real for the second statements	
	Contraction of the local data and the local data an		Thereit and a first set of the list of	1 Internation without and	
		Table Concerns	Construction of the second second	In 1 Contra Departure of	
and the second		And a second second	The local second second second second	Transford particular, difficultures	
and the second se	The second secon	and the second se	- Interference and the second	and a second sec	10
and the rest of the local division of the lo	Department with the line	and the second	The second second second		107
and and an	Tenned relations?		The second second second second second		
and a feature of the second	- man from the control of the		Transferrate Pallet		
intervence and a sector sector sector	- Particular Contraction		Construction and a second second		
and the second se	Taxas addresses		THE REPORT OF THE PARTY OF THE		
and a second s	The property of the local data		Constructions (180) 800		
and and a second s	and the second second		And		
and the second se			and the second		
a second s			the second se		
a statute (sectore)					
and the second se			and the second se		
and the second se			1000 S 2000		
C. and an other	-	100 C			
Comparison in the local data		Contra Contra			
The state of the s		A subject on			
These sectors in these		Automatics, and interest			
THE REPORT OF THE OWNER.	the second se	The second second second			
The second designed in the second sec	and the second s	The second second	and the second se		
Trainer, page (1997)	- State of the second s	the second second second			
	International Contract of the local data				
A search provide and the second					
Anna parate of the second	Carl Contraction of the Carl Carl Carl Carl Carl Carl Carl Carl	the second se			
A support of the latter of the	Internet states				
A page particular (with the same	- sale and other state				
A set prove of a set	Internet states				

Fi	le Hon	ne Inser	rt Draw	Page Layout F	ormulas Data Re	view View	Help		🖻 Share 🖓	Comme
G1		• E 2	x 🗸 :	fx AssayName						
	А		В	L I	K	Q	S	AB	AD	AH
1	assayID	publication	n_DOI	CoreComposition	CoatingComposition	OuterDiameter	OuterDiameter	l SurfaceArea	SurfaceAreaUnit	Shape
260	93	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
261	94	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
262	95	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
263	96	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
264	97	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
265	98	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
266	99	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
267	100	10.1080/1	7435390.20	copper(ii) oxide			nanometers	29	square meters/gram	spherica
268	101	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
269	102	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
270	103	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
271	104	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
272	105	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
273	106	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
274	107	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
275	108	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
276	109	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
277				copper(i) oxide	polyvinylpyrrolidone	48	nanometers		square meters/gram	
278				copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
279				copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	
280	113	10.1080/1	7435390.20	copper(i) oxide	polyvinylpyrrolidone	48	nanometers	20	square meters/gram	

# **APPROACH (cont.)**

# MySQL script to flatten the relational database NaKnowBase, pivoting on tables to place multiple concatenated entries for each experiment into one unique row:

1	CREATE OR REPLACE VIEW dev naknowbase.assay_all_vw AS CRUE		CREATE · OR · REPLACE · VIEW · dev_naknowbase . additive_vw · AS
2		2	📮 detail view.of.additive.self-joined.to.compute.rankuses.additive.rank.technique.with.CASE 眠 🖽
3	to separate rows into columns. The additive fields are concatenated into one field.	3	·to·separate·rows·into·columns.·The·additive·fields·are·concatenated·into·one·field.· 💽 🌆
4	the additive joins and case statement transform up to 16 rows into 16 columns field.	4	the additive joins and case statement transform up to 16 rows into sixteen columns field.
5	SELECT.ay.assayID,.ay.publication DOI,.ay.medium MediumID,.ay.medium publication DOI,.ay.material MaterialID,.ay. ORIN	5	·rank() ·is ·not ·a ·function ·until ·mysgl ·8.0. · so ·this ·uses ·the ·non-ggui ·outer ·join ·and · count ·to ·rank 🚱 🖽
6	material publication DOI, ay.assayType, ay.AssayName, mav.mediumDescription, @RMS	6	SELECT ad.medium_MediumID, ad.medium_publication_DOI, @RUS
7		7	count(ad2.idadditive)+1.idrank, CRUS
8		8	<pre></pre>
9	max (mav.additive02 num name amt unit) AS additive02 num name amt unit, CENT	9	
.0	max(rv.result01 num type dtl val approx unit uncert low hi) AS result01 num type dtl val approx unit uncert low hi, ORD	11	cannot-use-max.in.case.since.it.already.has.a.count CRM
1	······max(rv.result02 num type dtl val approx unit uncert low hi)·AS·result02 num type dtl val approx unit uncert low hi, OR III	12	□ · CASE · WHEN · count (ad2. idadditive) +1 · = · 1 · THEN · @RUE
2	max(pv.parameters01 num name numval unit nonnum) AS parameters01 num name numval unit nonnum, CRM	13	dev naknowbase.additive concat fun (ad.idadditive, ad.AdditiveAmount, ad.AdditiveUnit, ad.AdditiveName)
3	max (pv.parameters02 num name numval unit nonnum) AS parameters02 num name numval unit nonnum, CRM	14	- END AS additive01 num name amt unit, CRUS
4	······································	15	- CASE · WHEN · count (ad2.idadditive) +1 · = · 2 · THEN · CRUE
5	·····replace (pub.PubTitle, ·char(10), ·'·') ·PubTitle, ·pub.Journal, ·pub.year, ·pub. `First Author` ·AS · FirstAuthor, · CRUE	16	····dev_naknowbase.additive_concat_fun(ad.idadditive, ·ad.AdditiveAmount, ·ad.AdditiveUnit, ·ad.AdditiveName) @ @
6	·······pub.Volume, ·pub.Issue, ·pub.PageStart, ·pub.PageEnd, ·replace (pub.Keywords, ·char(10), · · · ) ·Keywords (R)	17	- END.ASadditive02_num_name_amt_unit@BB
7	PROM. dev naknowbase.assay.ay@BD	18	FROM dev_naknowbase.additive.ad@R#3
0	JOIN dev naknowbase.publication pubCRUS	19	LEFT.OUTER.JOIN.dev_naknowbase.additive.ad2@RDB
0	···ON·ay.publication DOI ·=·pub.DOI @RM3	20	··ON·ad.idadditive·>·ad2.idadditive
19		21	·AND·ad.medium_MediumID·=·ad2.medium_MediumID@303
20	JOIN.dev_naknowbase.medium_additive_vw.mav@RDB	22	<pre>•AND.ad.medium_publication_DOI.=.ad2.medium_publication_DOI@R000</pre>
1	•• ON•ay, medium_MediumID == mav.MediumIDORMS	23	this.group.by.returns.the.same.number.of.rows.as.without.group.by.It.is.needed.for.the.rank.count
2	AND.ay.medium_publication_DOI.=.mav.publication_DOI	24	GROUP·BY·ad.idadditive, ·ad.medium_MediumID, ·ad.medium_publication_DOI@RMG
23	result can have 0. to 18 rows per assay, ranked and concatenated	25	ORDER BY ad.medium_MediumID, ad.medium_publication_DOI, idrank, ad.idadditive;
24	LEFT.OUTER.JOIN.dev_naknowbase.result_vw.rv@RM	26 27	CREATE OR REPLACE VIEW dev_naknowbase.medium_additive_vw.AS@RD
25	••ON•ay,AssayID==rv,assay_AssayID@R00	28	SELECT.md.MediumID,.md.publication DOI,.md.MediumDescription, GRUS
26	AND.ay,publication_DOI.=.rv.assay_publication_DOI	29	max(additive01 num name amt unit) AS additive01 num name amt unit, GRUS
27	parameters.can.have.0.to.19.rows.per.assay,.ranked.and.concatenated	30	max (additive02 num name amt unit) AS additive02 num name amt unit
28	LEFT.OUTER.JOIN.dev_naknowbase.parameters_vw.pv@RM	31	FROM dev naknowbase.medium.md@RMS
29	··ON·ay.AssayID·=·pv.assay_AssayID@	32	LEFT.OUTER.JOIN.dev naknowbase.additive vw.adv@203
30	AND ay.publication_DOI.=.pv.assay_publication_DOI	33	··ON·md.MediumID·=·adv.mediumIDCRUS
31	GROUP.BY.ay.assayID,.ay.publication_DOI,.pub.DOI, @Reference.	34	·AND.md.publication DOI.=.adv.medium publication DOI
32	······mav.mediumID,·mcv.materialID,·mr.MolecularResultID,·mcv.MaterialFGID@RD	35	GROUP BY .md.MediumID, .md.publication DOI, .md.MediumDescription
33	ORDER.BY.ay.assayid,.ay.publication_DOI; CRUE	36	ORDER BY md.mediumid, md.publication DOI;

# **APPROACH (cont.)**

### Python code to interpret the multiple concatenated entries for each row into multiple columns:

```
@author: Wilson Melendez
import re
def split_additive_fields(df):
   Name
   split_additive_fields
   Description
   This function splits up the concatenated fields containing the additives.
    . . .
   # Extract column names
   column_names = list(df.columns)
   # Determine number of rows in data frame.
   nrow = len(df.index)
   # Process the additive fields
   additive_regex = re.compile(r'additive\d\d_num_name_amt_unit')
    list_additives = list(filter(additive_regex.match, column_names))
   num_additives = len(list_additives)
    try:
       for icol in range(0, num_additives):
            if (df[list_additives[icol]].isna().values.all() == True):
                continue
            for irow in range(0, nrow):
                if (df[list_additives[icol]].iloc[irow] == None):
                    continue
                else:
                   list_str = df[list_additives[icol]].iloc[irow].split(":")
                    # If additive name was not present, throw an exception.
                   if (list str[1] == ''):
                        error_message = "Name is missing for additive " + list_additives[icol] + " at row " + irow
                        raise ValueError(error_message)
                   # list_str[0] = number
                   # list_str[1] = name of additive
                   # list_str[2] = amount of additive (numeric value)
                   # list str[3] = unit of numeric value
                    strvalue = list_str[1].strip().lower() + ' additive_value'
                    strunits = list_str[1].strip().lower() + ' additive_unit'
                    if (list_str[2] != ''):
                        df.loc[irow, strvalue] = float(list_str[2])
                   else:
                        df.loc[irow, strvalue] = None
                    if (list_str[3] != ''):
                        df.loc[irow, strunits] = list_str[3]
                    else:
                        df.loc[irow, strunits] = None
    except ValueError as msg:
        error_message = msg + ", additive = " + list_additives[icol] + ", row = " + irow
        print(error_message)
   # Print message to console indicating completion of this function's task.
   print("Splitting of concatenated additive fields has completed.")
```