

WHATSNEXT: Guidance-enriched Exploratory Data Analysis with Interactive, Low-Code Notebooks

Chen Chen, Jane Hoffswell, Shunan Guo, Ryan Rossi,

Yeuk-Yin Chan, Fan Du, Eunyee Koh, Zhicheng Liu







Computation Notebooks are Popular for EDA

Data Science by using pandas liabrary

2]:	df = pd.DataF df.style	rame([i	[38.0, 2.0 ndex=pd.In plumns=pd.	, 18.0, 22.0 dex(['Tumour MultiIndex.f	, 21, np. (Positiv rom_produ	nan],[19, 43 me)', 'Non-Tu ct([['Decisi	9, 6, 45 mour (Ne on Tree	52, 226,232] egative)'], ', 'Regressi], name='Actual Label:'), on', 'Random'],['Tumour',	'Non-Tumour']], names=['Mo	del:', 'Predicted:']))
[2]:		Model:		Decision Tree		Regression		Random			
	Pr	edicted:	Tumour	Non-Tumour	Tumour	Non-Tumour	Tumour	Non-Tumour			
	Actu	I Label:									
	Tumour (F	ositive)	38.000000	2.000000	18.000000	22.000000	21	nan			
	Non-Tumour (N	gative)	19.000000	439.000000	6.000000	452.000000	226	232.000000			
[3]:	weather_df =	pd.Dat	aFrame(np. ind col	random.rand(ex=pd.date_r umns=["Tokyo	10,2)*5, ange(star ", "Beiji	t="2021-01-0 ng"])	1", peri	iods=10),			
	def rain_cond if v < 1. retur elif v < retur return "H	ition(75: n "Dry 2.75: n "Rain eavy R	/): n" ain"								
	def make_pret styler.se styler.fo styler.ba return st	ty(sty: t_capt: rmat(ri rmat_ii ckgrou yler	ler): ion("Weath ain_condit ndex(lambd nd_gradien	er Condition ion) a v: v.strft t(axis=None,	s") ime("%A") ∨min=1,) vmax=5, cmap	="YlGnBu	ı")			
3]:	T	okyo	Beijing								
	2021-01-01 3.62	1969 4	508378								
	2021-01-02 0.10	9078 3	062590								
	2021-01-03 0.49	3561 1	459806								
	2021-01-04 3.30	7517 0	808751								
	2021-01-05 3.70	9100 2	608226								
	2021-01-06 4.16	5616 0	519589								
	2021-01-07 4.69	5969 1	588812								
	2021-01-08 4.54	0735 0	477025								
	2021-01-09 3.14	9028 3	671971								
	2021-01-10 3.04	1754 0	944977								
4]:	weather_df.loc	["2021	-01-04":"2	.021-01-08"].	style.pip	e(make_prett	:y)				
4]:	Weather G	onditions									
		Tokyo	Beijing								
	Monday Hea	vy Rain	Dry								
	Tuesday Hea	vy Rain	Rain								
	Wednesday Hea	v Rain	Dry								
	incomestady inco										

- Multimodality
 - text, code, visualizations, and tables
- Flexibility
 - changes reflected in real-time



	Data Scien	ce by	using	pand	las liab	rary		
In [2]:	<pre>import pandas as pd import numpy as np import matplotlib as</pre>	s mpl						
In [2]:	df = pd.DataFrame([[ir cc df.style	[38.0, 2.0 ndex=pd.In plumns=pd.	, 18.0, 22.0 dex(['Tumour MultiIndex.f	, 21, np. (Positiv rom_produ	nan],[19, 43 e)', 'Non-Tu ct([['Decisi	9, 6, 45 mour (Ne on Tree'	2, 226,232]] gative)'], na , 'Regression	<pre>me='Actual Label:'), ', 'Random'],['Tumour', 'Non-Tumour']], names=['Model:', 'Predicted:']))</pre>
Out[2]:	Model:		Decision Tree		Regression		Random	
	Predicted:	Tumour	Non-Tumour	Tumour	Non-Tumour	Tumour	Non-Tumour	
	Actual Label:							
	Tumour (Positive)	38.000000	2.000000	18.000000	22.000000	21	nan	
	Non-Tumour (Negative)	19.000000	439.000000	6.000000	452.000000	226	232.000000	
In [3]:	<pre>weather_df = pd.Data def rain_condition(\ </pre>	aFrame(np. ind col	random.rand(ex=pd.date_r umns=["Tokyo	10,2)*5, ange(star ", "Beiji	t="2021-01-0 ng"])	1", peri	ods=10),	
	if v < 1.75: return "Dry" elif v < 2.75: return "Rain return "Heavy Ra	' n" ain"						
	<pre>def make_pretty(sty] styler.set_capti styler.format(ra styler.format_ir styler.backgrour return styler weather_df</pre>	ler): ion("Weath ain_condit ndex(lambd nd_gradien	er Condition ion) a v: v.strft t(axis=None,	ime("%A") vmin=1,) ∨max=5, cmap	="YlGnBu	")	
Out[3]:	Tokyo	Reijing						
	2021-01-01 3.621969 4.	508378						
	2021-01-02 0.109078 3.	062590						
	2021-01-03 0.493561 1.	459806						
	2021-01-04 3.307517 0.	808751						
	2021-01-05 3.709100 2.	608226						
	2021-01-06 4.165616 0.	519589						
	2021-01-07 4.695969 1.	588812						
	2021-01-08 4.540735 0.	477025						
	2021-01-09 3.149028 3.	671971						
	2021-01-10 3.041754 0.	944977						
In [4]:	weather df.loc["2021.	-01-04"+"2	021-01-08"1	style pir	e(make prett	v)		
0.000	Weather Conditions		or oo].		et and et a			
out[4]:	Tokyo	Beijing						
	Monday Heavy Rain	Dry						
	Tuesday Heavy Rain	Rain						
	Wednesday Heavy Rain	Dry						
	Thursday Heavy Rain	Dry						
	manaday meany Nam	0.9						

 Notebooks' <u>code-reliance</u> limits its usage by inexperienced programmers



	Data Scien	ce by	using	pand	las liab	rary				
]:	<pre>import pandas as pd import numpy as np import matplotlib as</pre>	; mpl								
[2]:	df = pd.DataFrame([[ir co	38.0, 2.0 dex=pd.In plumns=pd.), 18.0, 22.0 dex(['Tumour MultiIndex.f	, 21, np. (Positiv rom_produ	nan],[19, 439 e)', 'Non-Tur ct([['Decisio	9, 6, 45 mour (Ne on Tree'	2, 226,232]], gative)'], name , 'Regression'	<pre>='Actual Label:'), 'Random'],['Tumour',</pre>	'Non-Tumour']], names=['Mode	<pre>1:', 'Predicted:']))</pre>
	df.style									
[2]:	Model:		Decision Tree		Regression		Random			
)	Predicted:	Tumour	Non-Tumour	Tumour	Non-Tumour	Tumour	Non-Tumour			
	Actual Label:									
	Tumour (Positive)	38.000000	2.000000	18.000000	22.000000	21	nan			
	Non-Tumour (Negative)	19.000000	439.000000	6.000000	452.000000	226	232.000000			
[3]:	weather_df = pd.Data	aFrame(np. ind col	random.rand(lex=pd.date_r .umns=["Tokyo	10,2)*5, ange(star ", "Beiji	t="2021-01-0: ng"])	l", peri	ods=10),			
	<pre>def rain_condition(v if v < 1.75: return "Dry' elif v < 2.75: return "Rain"</pre>	/):								
	<pre>def make_pretty(sty] styler.set_capti styler.format(re styler.format_ir styler.backgrour return styler weather_df</pre>	ler): lon("Weath ain_condit ndex(lambd nd_gradien	er Condition ion) la v: v.strft t(axis=None,	s") ime("%A") vmin=1,) vmax=5, cmap	="YlGnBu	")			
:[3]:	Tokyo	Beijing								
	2021-01-01 3.621969 4	508378								
	2021-01-02 0.109078 3.	062590								
	2021-01-03 0.493561 1	459806								
	2021-01-04 3 207517 0	000751								
	2021-01-04 5.507517 0.	000751								
	2021-01-05 3.709100 2.	608226								
	2021-01-06 4.165616 0.	519589								
	2021-01-07 4.695969 1.	588812								
	2021-01-08 4.540735 0.	477025								
	2021-01-09 3.149028 3.	671971								
	2021-01-10 3.041754 0.	944977								
[4]:	weather_df.loc["2021	-01-04":"2	2021-01-08"].	style.pip	e(make_prett	y)				
141.	Weather Conditions									
	Tokyo	Beijing								
	Monday Heavy Rain	Dry								
	Tuesday Heavy Rain	Rain								
	Wednesday Heavy Rain	Day								
	Thursday Heavy Rain	Day								
	Inursday Heavy Rain	Dry								

- Notebooks' <u>code-reliance</u> limits its usage by inexperienced programmers
- Notebooks present <u>a single,</u>

interleaved thread, which may not capture the user's analysis flow



:]:	<pre>import pandas as p import numpy as np import matplotlib</pre>	as mpl								
2]:	df = pd.DataFrame([[38.0, 2.0 index=pd.Ir columns=pd.	0, 18.0, 22.0 ndex(['Tumour .MultiIndex.f	, 21, np.r (Positive	nan],[19, 439 e)', 'Non-Tum tt([['Decisio	0, 6, 45 nour (Ne on Tree'	2, 226,232]], gative)'], nam , 'Regression'	<pre>me='Actual Label:'), ', 'Random'],['Tumour',</pre>	'Non-Tumour']], names=['Model:', '	Predicted:']))
	df.style									
2]:	Mode	el:	Decision Tree		Regression		Random			
)	Predicte	d: Tumour	Non-Tumour	Tumour	Non-Tumour	Tumour	Non-Tumour			
	Actual Labe	el:								
	Tumour (Positiv	e) 38.000000	2.000000	18.000000	22.000000	21	nan			
	Non-Tumour (Negativ	e) 19.000000	439.000000	6.000000	452.000000	226	232.000000			
3]:	weather_df = pd.Da	ataFrame(np inc col	.random.rand(dex=pd.date_r Lumns=["Tokyo	10,2)*5, ange(start ", "Beijir	t="2021-01-01 ng"])	l", peri	ods=10),			
	def rain_condition if v < 1.75: return "Dr elif v < 2.75: return "Ra return "Heavy	n(v): ny" nin" Rain"								
	<pre>def make_pretty(st styler.set_cap styler.format(styler.format styler.backgro return styler weather_df</pre>	yler): ption("Weath (rain_condit index(lambo pund_gradier	ner Condition iion) da v: v.strft nt(axis=None,	s") ime("%A"); vmin=1, v) /max=5, cmap=	•"YlGnBu	-)			
1:	<pre>def make_pretty(st styler.set_cap styler.format(styler.format, styler.formaty styler.obackgro return styler weather_df Tokyo </pre>	:yler): tion("Weath rain_condit index(lambo und_gradier Beijing	ner Condition tion) Ja v: v.strft tt(axis=None,	s") ime("%A")) vmin=1, v) /max=5, cmap	•"YlGnBu	")			
	def make_pretty(st styler.set_cag styler.format styler.format styler.backgrr return styler weather_df Tokyo 2021-01-01 3621969	yler): tion("Weath rain_condit index(lambo pund_gradier Beijing 4508378	ner Condition tion) ia v: v.strft nt(axis=None,	s") ime("%A"); vmin=1, v) /max=5, cmap	-"YlGnBu	-)			
	def make_pretty(st styler.set_cag styler.formati styler.formati styler.backgro return styler weather_df Tokyo 2021-01-01 3.621969	<pre>syler): ption("Weath rain_condi index(lambo bund_gradier Beijing 4.508378 3.062590</pre>	her Condition iion) Ja v: v.strft ht(axis=None,	s") ime("%A")) vmin=1, v) /max=5, cmap:	∵"YlGnBu	")			
]: .	def make_pretty(st styler.set_cag styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-04.0 0.03551	<pre>syler): tion("Weath (rain_condit index(lamb bund_gradier Beijing 4.508378 3.062590 1.450805</pre>	her Condition iion) Ja v: v.strft ht(axis=None,	s") ime("%A"); vmin=1, v) /max=5, cmap=	="YlGnBu	-)			
]: .	def make_pretty(st styler.set_cag styler.formati styler.formati styler.backgro- return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-01-03 0.0493661	<pre>syler): tion("Weath rain_condit index(lambo bund_gradier Beijing 4.508378 3.062590 1.459806 0.000751</pre>	her Condition tion) Ja v: v.strft t(axis=None,	s") ime("%A"); vmin=1, v) /max=5, cmap:	="YlGnBu	")			
	def make_pretty(st styler.set_cag styler.formati styler.formati styler.backgrr return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-01-02 0.493561 2021-01-04 3.307517	tyler): tylen("Weath (rain_condit index(lambc und_gradier 4.508378 3.062590 1.459806 0.808751	her Condition tion) Ja v: v.strft t(axis=None,	s") ime("%A"); vmin=1, v) /max=5, cmap+	-"YlGnBu	-)			
]: .	def make_pretty(st styler.set_cay styler.formati styler.formati styler.beckgro return styler weather_df 2021-01-01 3.621969 2021-01-03 0.493561 2021-01-04 3.307517 2021-01-04 3.307517	tyler): tilon("Weath rain_condit index(lambe bund_gradier 4.508378 3.062590 1.459806 0.808751 2.608256	ner Condition iion) la v: v.strft nt(axis=None,	s") ime("%A")) vmin=1, v) max+5, cmap+	"YlGnBu	-)			
	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-0 3621969 2021-01-0 0.493561 2021-01-0 0.493561 2021-01-0 3.37517 2021-01-0 3.37517	tyler): tion("Weath rain_condit index(lambe bund_gradier Beijing 4.508376 3.062590 1.459806 0.808751 2.608226 0.519589	ner Condition ion) a v: v: strft t(axis=None,	s") vmin=1, v) max=5, cmap=	="YlGnBu	-)			
]:	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621969 2021-01-03 0.493561 2021-01-03 0.493561 2021-01-03 0.493561 2021-01-05 3.709100 2021-01-06 4.165616 2021-01-07 4.695969	yler): ption("Weath rain_condit index(lambu und_gradier	her Condition iion) a v: v.strft t(axis=None,	s") ime("%A")) vmin=1, v) max-5, cmap+	="YlGnBu	~)			
]: .	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-01-03 0.493561 2021-01-04 3.307517 2021-01-06 4.165616 2021-01-06 4.165616 2021-01-06 4.540735	ylen): tion("keath index(lambc und_gradier 4.508378 3.062590 1.459806 0.808751 2.608226 0.519589 1.588812 0.477025	her Condition Lion) a v: v.strft t(axis=None,	s") ime("%A")) vmin=1, v) max=5, cmap=	•"YlGnBu	")			
	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621969 2021-01-03 0.493561 2021-01-03 0.493561 2021-01-03 0.493561 2021-01-04 4.3307517 2021-01-05 3.709100 2021-01-06 4.165616 2021-01-07 4.693969 2021-01-08 4.540735 2021-01-09 3.149028	yler): tion("Weath index(lambc und_gradier 4.508378 3.062590 1.459806 0.808751 2.608226 0.519589 1.588812 0.477025 3.671971	ner Condition) Ja v: v.strft t(axis=None,	s") ime("%A") vmin=1, v) max-5, cmap-	-"YlGnBu	")			
]: .	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-01-03 0.493561 2021-01-03 0.493561 2021-01-04 3.307517 2021-01-05 3.709100 2021-01-05 3.709100 2021-01-06 4.165616 2021-01-07 4.693969 2021-01-08 4.540735 2021-01-09 3.149028 2021-01-03 3.041754	yler): tion("Weath rian_condition index(lambe bund_gradier 4.508378 3.062590 1.459806 0.808751 2.608226 0.519589 1.58812 0.477025 3.671971 0.944977	ner Condition) Ja v: v.strft t(axis=None,	s") ime("%A") vmin=1, v) mmax-5, cmap	"YlGnBu	")			
1:	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-01-03 0.493561 2021-01-03 0.493561 2021-01-04 3.307517 2021-01-05 3.709100 2021-01-06 4.165616 2021-01-07 4.695969 2021-01-07 4.695969 2021-01-08 4.540735 2021-01-09 3.149028 2021-01-09 3.149028	yler): tion("keath rein_condition index(lambe und_gradier Beijing 4.508376 3.062590 1.459806 0.808751 2.608226 0.519589 1.588812 0.477025 3.671971 0.944977 21-01-04":";	ner Condition ion) Ja v: v.strft t(axis=None,	s") ime("%A"); vmin=1, v) max-5, cmap- e(make_prett;	Y)	")			
	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgrd return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-01-03 0.493561 2021-01-04 3.307517 2021-01-04 3.307517 2021-01-05 3.709100 2021-01-05 4.165616 2021-01-06 4.165616 2021-01-07 4.695969 2021-01-08 4.540735 2021-01-08 3.440735 2021-01-08 4.540735 2021-01-08 4.54075 2021-01-08 4.54075 2021-08 4.54075 2021-08 4.54075 2021-08 4.54	yler): tion("Weath rein_condition index(lambe und_gradier 8eijing 4.508376 3.062590 1.459806 0.808751 1.459806 0.808751 0.808751 2.608226 0.519599 1.588812 0.477025 3.671971 0.944977 21-01-04":";";	ner Condition ion) Ja V: V.Strft t(axis=None, tt(axis=None, 2021-01-08"].	s") ime("%A")) vmin=1, v) max=5, cmap= e(make_prett;	rγ1GnBu ()	*)			
	def make_pretty(st styler.set_cas styler.formati styler.formati styler.beckgro return styler 2021-01-01 3.621969 2021-01-03 0.493561 2021-01-04 3.307517 2021-01-04 3.307517 2021-01-05 3.709100 2021-01-06 4.165616 2021-01-06 4.165616 2021-01-08 3.495969 2021-01-08 3.495959 2021-01-08 3.495959 2021-01-08 3.495959 2021-01-08 3.49575 2021-01-08 3.49575 2021-08 3.49575 2021-08 3.495755 2021-08 3.495755 2021-08 3.495755 2021-08 3.495	yler): tion("keath rein_condition index(lambe und_gradier Beijing 4.508378 3.062590 1.459806 0.808751 2.608226 0.808751 2.608226 0.808751 1.558812 0.477025 3.671971 0.944977 21-01-04":";";";";";";";";";";";";";";";";";";"	ner Condition ion) Ja V: V.Strft tt(axis=None, tt(axis=None, 2021-01-08"].	s") ime("%A"); vmin=1, v) max=5, cmap= e(make_prett;	r)	-)			
1 : . 1 : .	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621999 2021-01-02 0.109078 2021-01-02 0.499561 2021-01-03 0.499561 2021-01-04 3.307517 2021-01-04 3.307517 2021-01-05 3.709100 2021-01-04 4.165616 2021-01-07 4.469569 2021-01-08 4.540735 2021-01-08 4.540735 2021-01-08 3.449028 2021-01-10 3.041754 weather_df.loc("208: Toty Monday Heavy fissi	yler): ytion("Weath rein_condition index(lambe und_gradier 4.50875 3.062590 4.50875 1.459805 0.808751 2.608226 0.519589 1.58812 0.477025 3.671971 0.944977 21-01-04":"; ms 2 Beijing 0 Dry	<pre>ver Condition ion) Ja v: v.strft (axis=None, ver t(axis=None, ver t(x))) ver t(x)) ver</pre>	s") ime("%A"), v vmin-1, v) max-5, cmap+ e(make_prett;	r)	")			
	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgr return styler weather_df 2021-01-01 3.62190 2021-01-02 0.109078 2021-01-02 0.109078 2021-01-02 0.109078 2021-01-03 0.495561 2021-01-04 0.305171 2021-01-04 0.4659569 2021-01-04 4.65616 2021-01-04 4.659569 2021-01-05 0.70510 2021-01-05 0.70510 2021-01-06 4.659569 2021-01-08 4.540735 2021-01-09 0.149028 2021-01-08 0.4597868 Weather_df.loc["208 Weather_df.loc["208 Weather_conditional Weather_Conditional Weather_Conditional Weather_conditional Weather_df.loc["208 Weather_conditional Weat	yler): tion("keatt rein_condition eBeijing 4508378 3.062590 1.459806 0.808751 2.608226 0.519589 1.588812 0.477025 3.671971 0.944977 21-01-04":": bBeijing Dy Dy Rain	ner Condition Ja v: v.strft (axis=None,	s") ime("%4"), v wnin=1, v style.pip) max-5, cmap e(make_prett;	***Y1Gn8u	*)			
1= 1	def make_pretty(st styler.set_cas styler.formati styler.formati styler.backgro return styler weather_df 2021-01-01 3.621969 2021-01-02 0.109078 2021-01-03 0.493561 2021-01-03 0.493561 2021-01-04 0.435959 2021-01-05 3.709100 2021-01-05 3.709100 2021-01-04 4.659599 2021-01-04 4.65959 2021-01-05 3.709100 2021-01-05 3.709100000000000000000000000000000000000	Beijing 4.508376 3.062590 1.459806 0.8080751 2.608226 0.519589 1.588812 0.477025 3.671971 0.944977 21-01-04"; ";" Beijing Brain Brain Brain	ner Condition ion) Ja v: v.strft t(axis=None, tt(axis=None, 2021-01-08"].	s") ime("%4"), v wnin=1, v style.pip) max-5, cmap e(make_prett;	rYIGnBu r)	")			

- Notebooks' <u>code-reliance</u> limits its usage by inexperienced programmers
- Notebooks present <u>a single,</u>
 <u>interleaved thread</u>, which may not capture the user's analysis flow

much worse when you have many cells...



<pre>11 Supert products as pd Supert matpleliks as not Supert matpleliks as not import matpleliks as not if - pd.dstaframe([[5,6], 2,6], 31,6], 22,6], 21, ep.em], [19, 455, 6, 452, 252, 223]], columns pd.MultiIndex.from_product([] 'Decision Tree', 'Repression', 'Bandom'], 'Tumour', 'Non-Tumour']], names=('Nosel', 'Prestet' f.style</pre>												
import mutual is max if = gd DataFrace[[18:0, 2, 0, 18:0, 22.0, 21, np.nm],[19, 439, 5, 451, 226, 233]], indexed.Index[[Timour (Positive]], non-Tamour (Repression / Rendom)], Tame=-Kctual Lase1:'), column=pd.Nutlidex.from_product[[[10:01]] Mode Decides Tree Regression Rendom Predict: Tumour (Positive) Non-Tamour (Respective) Rendom Predict: Tumour (Positive) 200000 22 ren Non-Compositive Statumer (Tamour Non-Tumour Tumour Non-Tumour Non-Tumour Actual Label: '), indevad.Unit.c rage(start: "SDI-01-01'); (Tamour Non-Tumour Jumour Non-Tumour Actual Label: '), indevad.Unit.c rage(start: "SDI-01-01'); (Tamour '); (Tamour '): imp imp	port pandas as pd										
<pre>[2] of - p.d.btsfreme([183.e, 1.8, 1.8.d, 2.2.e, 21, pp.net][19, 498, 6, 452, 226, 323]], index.pd.fuel([Those(relative), "Won-lume(Regression, "Remover", "Non-Tumour", "Non-Tumour"]], names="{"Model:", "Predict ef.style</pre> Model Decision The Regression Random Predicted: Tumour Nen-Tumour Tumour Non-Tumour Non-Tumour Actual Label: Tumour OPeative0 9300000 2:000000 2:200000 2:2 2:200000 isologo 2:2 2:200000 2:2 2:200000 isologo 2:2 2:200000 2:2 2:200000 isologo 2:2 2:20000 isologo 2:20000	Tub	Jort matpiotiiD as	, mbr									
<pre>column=ud-Nultiladex.rom_product([['Decision Free', 'Regression', 'Rendom'];('Tumour', 'Ren-Tumour'], nemes-('Rodeli', 'Predict ef.usion Free Regression Random Predicts: Tumour Non-Tumour Tumour Non-Tumour Non-Tumour Actual table: Tumour (Restified 38.00000 2.00000 48.00000 2.21 nm Ren-Tumour (Restified 19.00000 2.800000 48.00000 2.22 22.00000) (setter_gf + pd.Ostframe(pp.random.rand(18.2)*5,</pre>	[2]: df	<pre>pd.DataFrame([[ir</pre>	38.0, 2.0 dex=pd.In), 18.0, 22.0 ndex(['Tumour	, 21, np. (Positiv	nan],[19, 43 e)', 'Non-Tu	9, 6, 45 mour (Ne	2, 226,232]] gative)'], n	, ame='Actual Label:'),			
Node: Decision Tre Repression Random Predicte: Tumour Non-Tumour Tumour Non-Tumour Tumour Non-Tumour Tumour Non-Tumour Tumour Non-Tumour	df.	style	olumns=pd.	MultiIndex.f	rom_produ	ct([['Decisi	on Tree'	, 'Regressio	n', 'Random'],['Tumour	', 'Non-Tumour']]], names=['Model:',	'Predicte
Predktd: Tumour Non-Tumour Tumour Non-Tumour Tumour Non-Tumour	[2]:	Model:		Decision Tree		Regression		Random				
Actual Lase: Tumour (Pealline) 3.00000 2.00000 21 nan Non-Tomour (Negline) 3.00000 2.00000 2.00000 2.00000 (a) westher_of = pd. Obstarse(np.rendom.rand(0,2)*5, indexod.det_range(itat2021-01-01), columns("Tokyn", "Beijing") periods=00, columns("Tokyn", "Beijing") periods=00, columns("Tokyn", "Beijing") def make_pretty(x1)rs); return "Baijn" return "Baijn		Predicted:	Tumour	Non-Tumour	Tumour	Non-Tumour	Tumour	Non-Tumour				
Tumour (Positive) 30:0000 20:0000 20:0000 21 nan Non-Tumour (Neglive) 9:00000 6:00000 22:00000 22:00000 22:00000 [3] weather_off = pd.lostarrame(np.rand(ng.))*5,		Actual Label:										
Nen-Tumour (Negstive 19.00000 499.00000 6.00000 420.00000 226 232.000000) uesther_df = pd.lotaframe(np.random.rand(10,2)'5, iddx:rdiadet_ange(tarr=2021-01-01", periods=10), claimes('Tokyo", "Beijing') uesther_df.print('); if f ('.1.75; returm "Bain" etim "Dyn" etif ('.2.75; returm "Bain" returm "Bain" returm "Bain" styler.formating(Nether Conditions") styler.formating(Nether Condins") styler.formating(Nether Conditions") styler.fo		Tumour (Positive)	38.000000	2.000000	18.000000	22.000000	21	nan				
<pre>[3] westher_df = pd.Dataframe(np.random.rand(18,2)*5,</pre>	Non	n-Tumour (Negative)	19.000000	439.000000	6.000000	452.000000	226	232.000000				
def rain_condition(v); if v < 1.75: if v < 1.75: if v < 1.75: returm "Bain" returm "Teavy Bain" def make_pretty(styler): styler.set_conditions") styler.set_condition styler.beckground_gradient(axis=None, wmin-1, vmax=5, cmap=""IGnBu") returm styler weather_df 2021-01-02 049078 2021-01-02 049078 2021-01-02 049078 2021-01-02 049078 2021-01-04 320757 2021-01-02 049078 2021-01-03 240920 2021-01-04 4307571 2021-01-04 4307571 2021-01-05 340202 2021-01-04 454075 2021-01-04 454075 2021-01-05 344020 2021-01-05 344070 weather_df.loc("2021-01-04-04":"2021-01-08"].style.pipe(make_pretty) weather_df.loc("2021-01-04-04":"2021-01-08"].style.pipe(make_pretty) weather_df.loc("2021-01-04-04":"2021-01-08"].style.pipe(make_pretty) weather_df.loc("2021-01-04-04":"2021-01-08"].style.pipe(make_pretty) <th>[3]: wea</th> <th>ather_df = pd.Data</th> <th>aFrame(np. ind col</th> <th>.random.rand(lex=pd.date_r Lumns=["Tokyo</th> <th>10,2)*5, ange(star ", "Beiji</th> <th>t="2021-01-0 ng"])</th> <th>1", peri</th> <th>ods=10),</th> <th></th> <th></th> <th></th> <th></th>	[3]: wea	ather_df = pd.Data	aFrame(np. ind col	.random.rand(lex=pd.date_r Lumns=["Tokyo	10,2)*5, ange(star ", "Beiji	t="2021-01-0 ng"])	1", peri	ods=10),				
<pre>true in recey much def make_continue. styler.format(rink_continue) st</pre>	def	<pre>f rain_condition(v if v < 1.75: return "Dry" elif v < 2.75: return "Rain return "Heave Rain"</pre>	/):									
Image (e) Takyo Bajing 2021-01-01 3.621969 4.508376 2021-01-02 0.09376 3.662599 2021-01-03 0.93561 1.459006 2021-01-04 3.007570 0.662590 2021-01-05 3.709100 2.608226 2021-01-06 4.56576 0.519589 2021-01-07 4.69599 1.558512 2021-01-08 4.54075 0.470725 2021-01-09 3.149028 3.671971 2021-01-09 3.041754 0.944977 412 iweather_df.loc("2021-01-04";"2021-01-08").style.pipe(make_pretty) 413: iweather_df.loc("2021-01-04";"2021-01-08").style.pipe(make_pretty) 413: iweather_df.loc("2021-01-04";"2021-01-08").style.pipe(make_pretty) 413: iweather_df.loc("2021-01-04";"2021-01-08").style.pipe(make_pretty) 414: iweather_df.loc("2021-01-04";"2021-01-08").style.pipe(make_pretty) 415: iweather_df.loc("2021-01-04";"2021-01-08").style.pipe(make_pretty) 416: iweather_df.loc("2021-01-08").style.pipe(make_pretty) 417: iweather_df.loc("2021-01-08").style.pipe(make_pretty)	def	<pre>make_pretty(styl styler.set_capti stylep_format(n)</pre>	ler): lon("Weath	er Condition	- ** >							
2021-01-01 3.621969 4.508376 2021-01-02 0.09078 3.062599 2021-01-03 0.495561 1.459506 2021-01-04 3.07517 0.808751 2021-01-05 3.709100 2.608226 2021-01-06 4.165616 0.519589 2021-01-07 4.65616 0.519589 2021-01-08 4.540735 0.4477025 2021-01-09 3.149028 3.617971 2021-01-04 0.944977 4.94497 412 weather_dcf.loc("2021-01-04";"2021-01-08"].style.pipe(make_pretty) 413: Weather Conditions Tokyo Beljing Monday Heavy Rain Dry Tuesday Heavy Rain Dry		styler.formatin styler.backgrour return styler	ain_condit ndex(lambd nd_gradien	tion) Wa v: v.strft t(axis=None,	s") ime("%A") vmin=1,) vmax=5, cmap	="YlGnBu	-)				
2021-01-02 0.103078 3.062390 2021-01-03 0.403581 1.85906 2021-01-04 3.07517 0.805751 2021-01-05 3.709100 2.608226 2021-01-05 4.16516 0.519589 2021-01-06 4.56975 1.85906 2021-01-06 4.56976 0.519589 2021-01-06 4.56975 0.477025 2021-01-08 5.671971 2.0044977 2021-01-10 3.041754 0.944977 4]: weather_df.loc("2021-01-064";"2021-01-068").style.pipe(make_pretty) 4]: Weather_Conditions Tokyo Beijing Monday Heavy Ban Dry Tuesdy Heavy Ban Dry Wednesday Heavy Ban Dry	wea	styler.format_in styler.backgrour return styler ather_df	hin_condit hdex(lambd hd_gradien	ion) Wa v: v.strft t(axis=None,	s") ime("%A") vmin=1,) vmax=5, cmap	="YlGnBu	")				
2021-01-02 0.109.07 3.002751 2021-01-03 3.079100 2.686226 2021-01-05 3.079100 2.686226 2021-01-06 4.165616 0.519589 2021-01-07 4.699999 1.588812 2021-01-08 4.540735 0.477025 2021-01-08 3.149028 3.671971 2021-01-10 3.041774 0.944977 41: weather_df-loc("2021-01-04":"2021-01-08"].style.pipe(make_pretty) 41: Tokyo Beijing Vedmesday Heavy Bain Day Vedmesday Heavy Bain Day	3]:	styler.format_ir styler.backgrour return styler sther_df	ain_condit ndex(lambd nd_gradien	ion) Wa v: v.strft t(axis=None,	s") ime("%A") vmin=1,) vmax=5, cmap	="YlGnBu	-)				
2021-01-03 3037517 0.003751 2021-01-05 3.703100 2.608226 2021-01-07 4.65516 0.519589 2021-01-07 4.65596 1.585812 2021-01-09 3.149028 3.671971 2021-01-09 3.149028 3.671971 2021-01-08 5.5528 2021-01-09 5.	3]: 2021	styler.format_in styler.backgrour return styler ather_df I-01-01 3.621969 4.	ain_condit dex(lambd d_gradien Beijing 508378	:ion) la v: v.strft t(axis=None,	s") ime("%A") vmin=1,) vmax=5, cmap	="YlGnBu	=)				
2021-01-05 3.70910 2.608226 2021-01-05 4.165616 0.519589 2021-01-06 4.540735 0.477025 2021-01-09 3.149028 3.671971 2021-01-10 3.041754 0.944977 4]: weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty) 4]: Weather_Conditions Tokyo Beijing Monday Heavy Ban Venesday Heavy Ban Weathersday Heavy Ban Venesday Heavy Ban	wea 3]: 2021 2021 2021	styler.format_ir styler.backgrour return styler ther_df 1-01-01 3.621969 4.	ain_condit dex(lambd d_gradien 508378 062590	:ion) Wa v: v.strft t(axis=None,	s") ime("%A") vmin=1,) vmax=5, cmap	="YlGnBu	-)				
2021-01-05 3/03100 2x86226 2021-01-06 4.165616 0.519599 2021-01-07 4.695969 1.588812 2021-01-08 4.54775 0.477025 2021-01-08 3.45775 0.547027 2021-01-08 3.45775 0.547027 2021-01-10 3.041754 0.944977 2021-01-10 3.041754 0.944977 4]: weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty) 4]: Weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty) 4]: Weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty) 4]: Weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty) 4]: Weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty) 4]: Weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty) 4]: Weather_df.loc["2021-01-04":"2021-01-08"].style.pipe(make_pretty)	wea 3]: 2021 2021 2021 2021	styler.format_ir styler.backgrour return styler ther_df 1-01-01 3.621969 4. 1-01-02 0.109078 3. 1-01-03 0.493561 1.	ain_condit dex(lambd nd_gradien 508378 062590 459806	:ion) Wa v: v.strft t(axis=None,	s") vmin=1,) vmax=5, cmap	="YlGnBu	-)				
2021-01-00 4.165516 0.519509 2021-01-00 4.659560 1.588812 2021-01-00 4.549735 0.477025 2021-01-00 3.041754 0.944977 2021-01-10 3.041754 0.944977 2021-01-00 5.10c["2021-01-04";"2021-01-08"].style.pipe(make_pretty) Vember_of.loc["2021-01-04";"2021-01-08"].style.pipe(make_pretty) Vember_of.loc["2021-01-04";"2021-01-08"].style.	3]: 2021 2021 2021 2021 2021 2021	styler.format_ir styler.backgrour return styler ther_df -01-01 3.621969 4. -01-02 0.109078 3. -01-03 0.493561 1. -01-04 3.3075010 2.	in_condit idex(lambd idex(lambd idex(lambd idex(lambd) 508378 062590 459806 808751 608255	:ion) la v: v.strft t(axis=None,	s') ime("%A") vmin=1, ') vmax=5, cmap	="YlGnBu	-)				
2021-01-00 4.540735 2021-01-08 4.540735 2021-01-09 3.149028 3.671971 2021-01-10 3.041754 0.944977 401 Weather_def.loc("2021-01-04"; "2021-01-08").style.pipe(make_pretty) 41: Weather Conditions Tokyo Beijing Monday Heavy Ran Dry Tuesday Heavy Ran Dry Wednesday Heavy Ran Dry	wea 3]; 2021 2021 2021 2021 2021 2021 2021	styler.format_ir styler.format_ir styler.backgrour return styler 101-01 3.621969 4. -01-02 0.109078 3. -01-03 0.493561 1. -01-03 0.493561 1. -01-04 3.307517 0.	in_condit idex(lambd d_gradien 508378 508378 508378 508378 508375 508375 508375 508226 508226	:ion) la v: v.strft t(axis=None,	s') ime("%A") vmin=1, ') vmax=5, cmap	="YlGnBu	-)				
2021-01-09 3.149025 Us47/VU23 2021-01-09 3.149025 3.671971 2021-01-10 3.041754 0.944977 4]: Weather_def.loc("2021-01-04": "2021-01-08").style.pipe(make_pretty) 4]: Weather Conditions Tokyo Beijing Monday Heavy Ban Dry Tuesday Heavy Ban Dry	wea 3]; 2021 2021 2021 2021 2021 2021 2021 202	Styler.format_in styler.format_in styler.format_in styler.backgrour return styler 1-01-01 3.621969 4. 1-01-02 0.109078 3. 1-01-03 0.493561 1. 1-01-04 3.307517 0. -01-05 3.709100 2.	in_condit idex(lambd d_gradien 508378 062590 459806 808751 608226 519589	:ion) la v: v.strft t(axis=None,	s') ime("%A") vmin=1, ') vmax=5, cmap	="YlGnBu	*)				
2021-01-00 3/241754 0.944977 2021-01-10 3/241754 0.944977 41: weather_df.loc("2021-01-04":"2021-01-08"].style.pipe(make_pretty) 42: Weather_conditions Tokyo Beijing Monday Heavy Rain Dry Tuesday Heavy Rain Dry Heavy Rain Dry	2021 2021 2021 2021 2021 2021 2021 2021	Styler.formatin styler.formatin styler.backgrour return styler -01-01 3.621969 4. -01-02 0.109078 3. -01-03 0.493561 1. -01-04 3.307517 0. -01-05 3.709100 2. -01-06 4.165616 0. -01-07 4.695969 1.	in_condit idex(lambd idex(lambd id_gradien 508378 062590 459006 808751 608226 519589 588812	ion) la vi v.strft t(axis=None,	s') ime("%A") vmin=1, ') vmax=5, cmap	="YlGnBu	")				
4): weather_df.loc["2021-01-04";"2021-01-08"].style.pipe(make_pretty) 4): Weather Conditions Tokyo Beijing Tokyo Beijing Monday Heavy Ran Dry Tuesday Heavy Ran Ran Wednesday Heavy Ran Dry	2021 2021 2021 2021 2021 2021 2021 2021	styler.hematis styler.backgrou return styler -01-01 3.621969 4. -01-02 0.03076 3. -01-03 0.03076 3. -01-03 0.435561 1. -01-04 3.307517 0. -01-05 3.307517 0. -01-05 3.307517 0. -01-05 4.155616 0. -01-07 4.635969 1.	in_condit idex(lambd d_gradien 508378 062590 459006 808751 608226 519589 588812 477025	ion) la v: v.strft t(axis=None,	s') ime("%A") vmin=1,:) wmax=5, cmap	="YlGnBu	")				
4]: weather_df.Joc("2021-01-04";"2021-01-08"].style.pipe(make_pretty) 4]: Weather Conditions Tokyo Beijing Monday Heavy Ran Dry Tuesday Heavy Ran Dry	wea 2021 2021 2021 2021 2021 2021 2021 202	Styler.format_ir styler.format_ir styler.format_ir styler.format_ir styler.format_ir styler.format_ir styler.format_ir -01-01 3.621969 -01-02 0.109078 -01-03 0.493561 -01-04 3.075170 -01-05 3.709100 -01-05 3.709100 -01-06 4.165616 0. -01-08 4.540735 0. -01-09 1.40208 1.40208 0.101-09 0.140208 0.140208	hin_condit dex(lambda) dd_gradien 508378 062590 459806 808751 608226 519589 588812 477025 671971 944977	ion) la v: v.strft t(axis=None,	s') ime("%A") vmin=1,) wmax=5, cmap	="YlGnBu	-)				
4]: Westher Conditions Tokyo Beijing Monday Heavy Rain Dry Tuesday Heavy Rain Rain Wednesday Heavy Rain Dry	vea 2021 2021 2021 2021 2021 2021 2021 202	Styler.format_ir styler.fo	iin_condit dex(lambda) dd_gradien 508378 062590 459806 808751 608226 519589 588812 477025 671971 944977	ion) la v: v.strft t(axis=None,	s) vmin=1, ·) wmax=5, cmap	="YlGnBu	-)				
Image: Torkyo Beijing Monday Heavy Rain Dry Dry Tuesday Heavy Rain Wednesday Heavy Rain	3]: 2021 2021 2021 2021 2021 2021 2021 202	Styler.beckgroup styler.beckgroup styler.beckgroup return styler return styler 101-01 3621969 -01-02 0.03076 -01-03 3.037517 -01-04 3.037517 -01-05 3.037517 -01-06 4.165616 -01-07 4.693560 -01-08 4.540735 -01-08 4.540735 -01-03 0.041754 -01-04 3.041754	<pre>in_condit dex(lambda) ddgradien Bailog 508378 062590 459806 808751 608226 519589 588812 477025 671971 944977 -01-04":"2</pre>	ion) la v: v: strft t(axis=None, t(axis=None, 2021-01-08"].	style.pip) wmax-5, cmap	="YlGnBu	-)				
Monday Heavy Rain Dry Tuesday Heavy Rain Rain Wednesday Heavy Rain Dry	<pre>wea 3]: 2021 2021 2021 2021 2021 2021 2021 4]: weat 4]:</pre>	30 Jan Jan Styler Normating styler Normating styler Normating styler Normating styler Normating Indiana Styler Indiana Styler <td>tin_condit ddex(lambdac) bailog 508378 062590 459806 608226 519589 588812 477025 671971 944977 -01-04":"2</td> <td>ion) Ha v: v: Strft Ha v: v: Strft Av: v: v: strft Av: strft</td> <td>s) vmin=1, ·</td> <td>) wmax-5, cmap e(make_prett</td> <td>s="YlGnBu y)</td> <td>-)</td> <td></td> <td></td> <td></td> <td></td>	tin_condit ddex(lambdac) bailog 508378 062590 459806 608226 519589 588812 477025 671971 944977 -01-04":"2	ion) Ha v: v: Strft Ha v: v: Strft Av: v: v: strft Av: strft	s) vmin=1, ·) wmax-5, cmap e(make_prett	s="YlGnBu y)	-)				
Tuesday Heavy Rain Rain Wednesday Heavy Rain Dry	<pre>wea 3]: 2021 2021 2021 2021 2021 2021 2021 4]: weat 4]:</pre>	Styler Onimat Ir styler backgrour styler backgrour return styler -01-01 3621969 -01-02 0.109078 -01-03 307517 -01-04 3307517 -01-05 3709100 -01-06 4.65616 -01-07 4.695969 -01-08 4.540735 -01-01 3.44754 -01-01 3.041754 -01-01 3.041754	tin_condit ddex(lambdac) 508378 062590 459606 808751 608226 519589 588812 477025 671971 944977 -01-04":"2 Beijing	ion) Id v: v:Strft H(axis=None, t(axis=None, t(axis=None, 2021-01-08"].	style.pip) wmax-5, cmap e(make_prett	="YlGnBu y)	-)				
Wednesday Heavy Rain Dry	з]: 2021 2021 2021 2021 2021 2021 2021 202	Styler. format_f styler. format_f styler. backgrour return styler return styler rolool 3.621969 -01-01 3.621969 -01-02 0.109078 -01-03 0.493561 -01-04 3.07517 -01-05 3.079100 -01-04 1.601-04 1.601-05 4.560755 -01-07 4.540755 -01-08 4.540755 -01-07 4.540755 -01-07 4.540755 -01-07 4.540755 -01-07 4.540756 -01-08 4.540757 -01-09 3.149028 -01-01 4.540754 Cher_dt.loc: -01-02 -01-03 -01-04 -01-05 -01-05 -01-07	tin_condit ddex(lambdac) 508378 508378 5082590 459806 808751 608226 519589 588812 477025 671971 944977 -01-04":"2 Beijing Dry	ion) Ha v: v:strft H(axis=None, t(axis=None, t(axis=None, 2021-01-08"].	s) vmin=1, style.pip) wmax=5, cmap	="YlGnBu y)	-)				
	weat 2021 2021 2021 2021 2021 2021 2021 202	styler.format_ir styler.format_ir styler.format_ir styler.format_ir styler.format_ir styler.format_ir styler.format_ir styler.format_ir lol-01 styler.format_ir lol-02 lol-03 styler.format_ir lol-04 307517 lol-04 styler.format_ir lol-04 307517 lol-04 styler.format_ir lol-04 styler.format_ir lol-04 styler.format_ir lol-04 styler.format_ir lol-05 styler.format_ir lol-04 <	tin_condit ddex(lambdex) 508378 062590 459806 808751 608226 519589 588812 477025 671971 944977 -01-04":"2 Beijing Dry Rain	ion) Ha v: v:strft H(axis=None, t(axis=None, t(axis=None, 2021-01-08"].	s-) ime("\$A") vwin=1, ·) wmax=5, cmap e(make_prett	="Y1GnBu	-)				

- Notebooks' <u>code-reliance</u> limits its usage by inexperienced programmers
- Notebooks present <u>a single</u>, <u>interleaved thread</u>, which may not

capture the user's analysis flow

We aim at an interactive notebook framework to

support efficient low-code exploratory data analysis



Design Goals

- Notebooks' **<u>code-reliance</u>** limits its usage by inexperienced programmers
 - **DG1: Low-code:** support users with varying levels of programming expertise.
 - **DG2: Insight-driven:** help quickly synthesize compound data insights.



Design Goals

- Notebooks' **<u>code-reliance</u>** limits its usage by inexperienced programmers
 - **DG1: Low-code:** support users with varying levels of programming expertise.
 - **DG2: Insight-driven:** help quickly synthesize compound data insights.
- Notebooks present **a single, interleaved thread**, which may not capture the user's analysis flow
 - **DG3: History:** help recall and navigate efficiently with visual cues and interactions.
 - **DG4: Structure:** reveal the analytic dependencies between cells.



WhatsNext: Guidance-enriched EDA with Interactive, Low-Code Notebooks

We use the following car sales dataset to present a usage scenario of WhatsNext.

Model	MPG	Cylinders	Displacement	Horsepower	Weight	Acceleration	Year	Origin
volkswagen 113	26	4	97	46	1835	20.5	70	Europe
volkswagen supe	26	4	97	46	1950	21	73	Europe
volkswagen rabb	43.1	4	90	48	1985	21.5	78	Europe
vw rabbit c (dies	44.3	4	90	48	2085	21.7	80	Europe
vw dasher (diese	43.4	4	90	48	2335	23.7	80	Europe
fiat 128	29	4	68	49	1867	19.5	73	Europe
toyota corona	31	4	76	52	1649	16.5	74	Japan
chevrolet chevet	29	4	85	52	2035	22.2	76	US
mazda glc delux	32.8	4	78	52	1985	19.4	78	Japan

https://goo.gl/9G1egz





DG1: Low-code: support users with varying levels of programming expertise.





DG2: Insight-driven: help quickly synthesize compound data insights.

DG4: Structure: reveal the analytic dependencies between cells.

DG3: History: help recall and navigate efficiently with visual cues and interactions.

DG3: History: help recall and navigate efficiently with visual cues and interactions.

DG4: Structure: reveal the analytic dependencies between cells.

Hover: to display a screenshot tooltip **Click:** go to the corresponding cell

Summarization

We introduce WHATSNEXT, an interactive notebook environment for low-code data exploration that

- augments a standard notebook cell with a <u>no-code interaction panel</u> showing recommended follow-up analysis questions or actions;
- utilize a set of <u>insight-driven heuristics</u> to synthesize follow-up questions to help reveal / explore / integrate data insights;
- visualizes the <u>analysis hierarchy</u> to help users trace the history of diverging analysis threads.

- Improve connections between the notebook and analysis structure.
 - More <u>annotations</u> to show hidden features, such as insight type / recommendation type / instructions...

- Improve connections between the notebook and analysis structure.
 - More <u>annotations</u> to show hidden features, such as insight type / recommendation type / instructions...
- Support user-customized EDA trajectories.
 - Support **<u>user-specified analysis</u>** intents
 - Maintain ecological synchronization

- Improve connections between the notebook and analysis structure.
 - More <u>annotations</u> to show hidden features, such as insight type / recommendation type / instructions...
- Support user-customized EDA trajectories.
 - Support **<u>user-specified analysis</u>** intents
 - Maintain ecological synchronization
- Support thread decomposition, switching, and saving.

- Improve connections between the notebook and analysis structure.
 - More <u>annotations</u> to show hidden features, such as insight type / recommendation type / instructions...
- Support user-customized EDA trajectories.
 - Support **user-specified analysis** intents
 - Maintain ecological synchronization
- Support thread decomposition, switching, and saving.
- Evaluate **<u>usability</u>** through a comparative study

WHATSNEXT: Guidance-enriched Exploratory Data Analysis with Interactive, Low-Code Notebooks

