`sklearn_talk.dataiku`.LeaveNoOneOut

Léo Dreyfus-Schmidt & Samuel Ronsin

LeaveNoOneOut

Machine Learning Accessible to *Everybody* ?

scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts

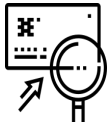
What about the other (non-Python-literate) *Everybody* ?

LeaveNoOneOut

Why make Machine Learning Accessible to Everybody ?

Expansion of ML applications

Predictive Maintenance



Fraud detection

Pricing

Churn Prediction

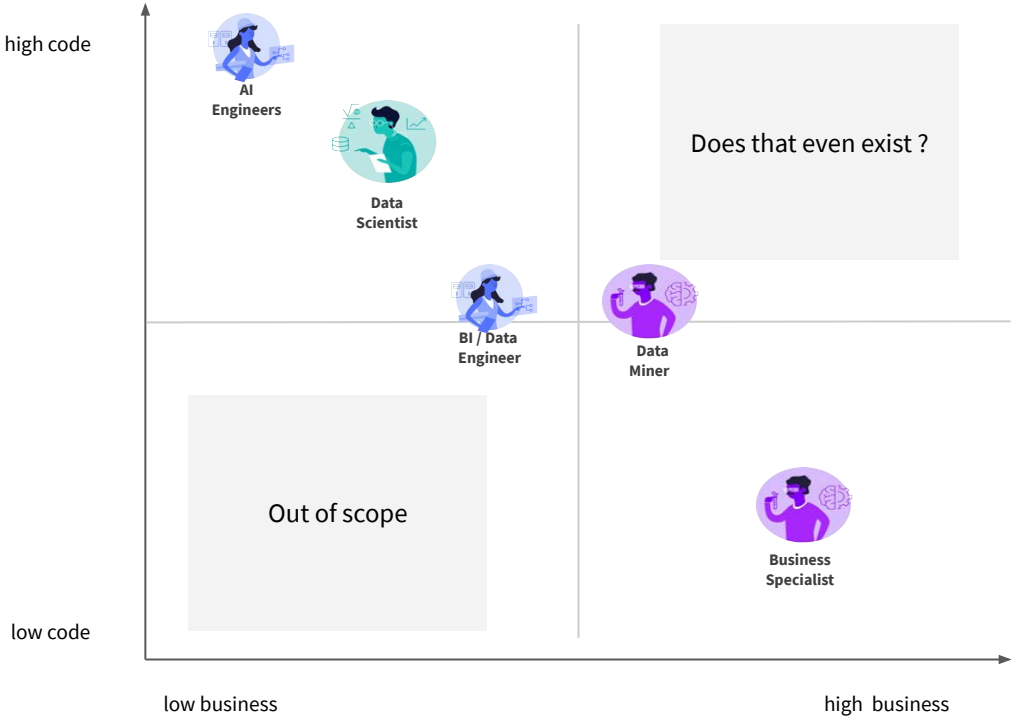
LeaveNoOneOut

Why make Machine Learning Accessible to Everybody ?

Expansion of ML applications

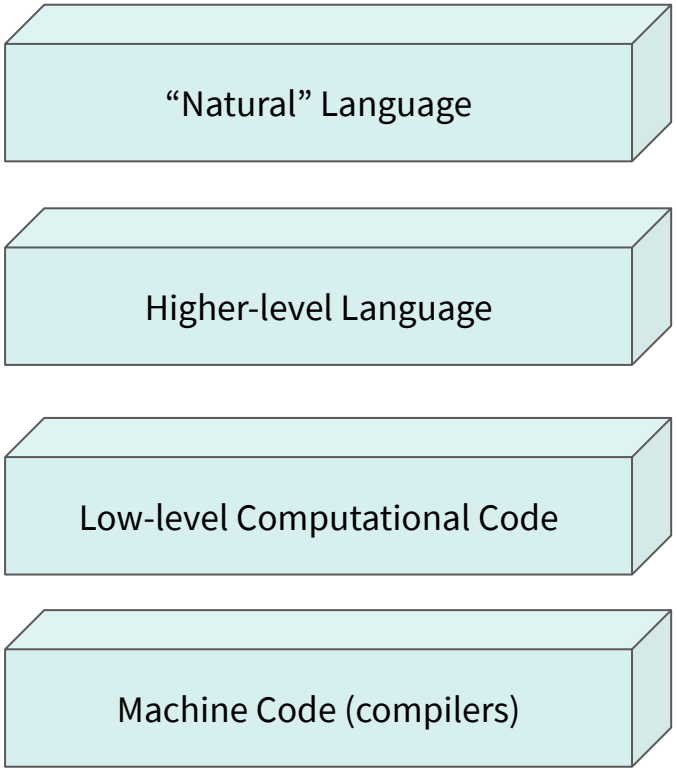
LeaveNoOneOut

Not Just ML experts



LeaveNoOneOut

ML Stack of Babel



`sklearn_talk.dataiku` Common Ground

LeaveNoOneOut

Common Grounds Abstraction

Simplicity

Universality

Empowerment

LeaveNoOneOut

Story 1: Building ML Pipelines Together

Business Analyst

Data Scientist

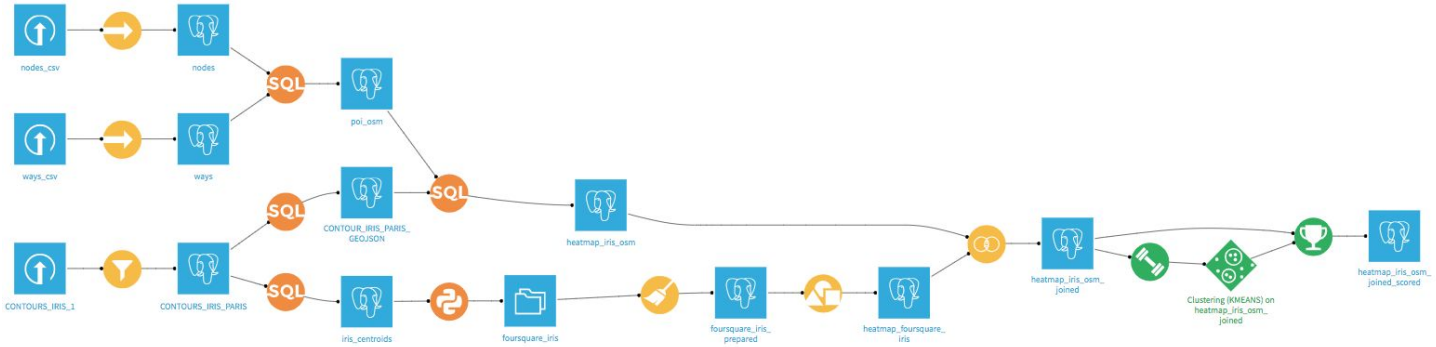
Business Analyst knows about the use case and the data

Data Scientist knows about ETL and ML

They need to build a ML model that optimizes a business metric

LeaveNoOneOut

Story 1: Building ML Pipelines Together



LeaveNoOneOut

Story 1: Building ML Pipelines with a *Clickodrome*

Customer Lifetime Value | Design node

Predict revenue v1 | Summary | Script | Charts | Models | DEPLOY SCRIPT | ACTIONS

Viewing design sample

10000 rows, 12 cols

DISPLAY

10000 matching rows

ip	web_ip_country	web_ip_geopoint	web_pages_visited	web_camp...	age	crm_price_first_item_purchased	crm_gender	crm_reven...	crm_value	join_GDP_cap
ss	Country	GeoPoint	Decimal	Boolean	Integer	Decimal	Gender	Integer		
1.56.200	United States	POINT(-97.822 37.751)	2.0	false	32	22.0	M	171		
1.243	Spain	POINT(-3.7635 40.3272)	6.0	false	40	44.0	M	156		
3.195.184	China	POINT(116.3883 39.9289)	5.0	true	23	28.0	F	130		
143.64	United States	POINT(-97.822 37.751)	6.0	false	43	22.0	M	165		
27.107	Spain	POINT(-5.5406 39.8916)	2.0	false	35	42.0	F	157		
5.37.70	United States	POINT(-97.822 37.751)	4.0	false	119	22.0	M	151		
221.52	South Africa	POINT(28.0583 -26.2309)	3.0	false	50	10.0	F	73		
92.61	Japan	POINT(139.69 35.69)	9.0	false	54	28.0	F	196		
3.50.102	United States	POINT(-87.1554 37.7513)	4.0	false	119	44.0	M	176		
3.10.174	United States	POINT(-97.822 37.751)	5.0	false	33	117.0	F	360	high	51704
146.4	Brazil	POINT(-43.2192 -22.8305)	5.0	false	24	15.5	F	77	low	11747
87.12	United States	POINT(-77.7417 38.7153)	8.0	false	27	44.0	F	255	high	51704
3.168.234	Brazil	POINT(-43.3307 -22.9201)	3.0	false	65	22.0	F	99	low	11747
193.74	United States	POINT(-78.8895 42.9167)	5.0	false	36	57.0	F	283	high	51704
.230	Montenegro	POINT(19 42)	6.0	false	31	57.0	F	150	low	11610
15.160	United Kingdom	POINT(-0.1224 51.4964)	1.0	true	33	15.5	F	115	low	36569
196.70	United States	POINT(-97.822 37.751)	8.0	false	28	44.0	M	224	high	51704
128.214	Germany	POINT(7.8378 49.8484)	5.0	false	36	44.0	M	188	high	38666
138.230	United States	POINT(-86.2379 41.7002)	9.0	false	51	22.0	M	188	high	51704

Delete
Rename
Move
Analyze...
Edit column metadata...
Remove rows with no value
Fill empty rows with...
More actions
Filter
Sort
Color column by value
Create Prediction model...

Visual ML powered by scikit-learn

Story 1: Building ML Pipelines with a *Clickodrome*

Visual ML = articulation of functions and objects taken from scikit-learn:

- Many `sklearn.feature_extraction.*Vectorizer`
- Many `sklearn.*{Classifier,Regressor}`
- Many `sklearn.model_selection.*{Split,KFold}`

  working around `GridSearchCV` to add live visual feedback
- Many `sklearn.metrics.*{score,error,loss}`

LeaveNoOneOut

Story 1: Building ML Pipelines with a *Clickodrome*

Pushing towards ML Best Practices

- Proper Train/Test split with metrics always computed on Test
- Automatic Handling of imbalanced data through `class_weight`

...

The AUC (Area Under the Curve) for this model is `1.000` , which is **...too good to be true?**.

LeaveNoOneOut

Story 2: Deploy ML Models to Production

Data Engineer

Data Scientist

Data Scientist needs to retrain periodically its ML models

Data Engineer needs to deploy ML models to a production REST API

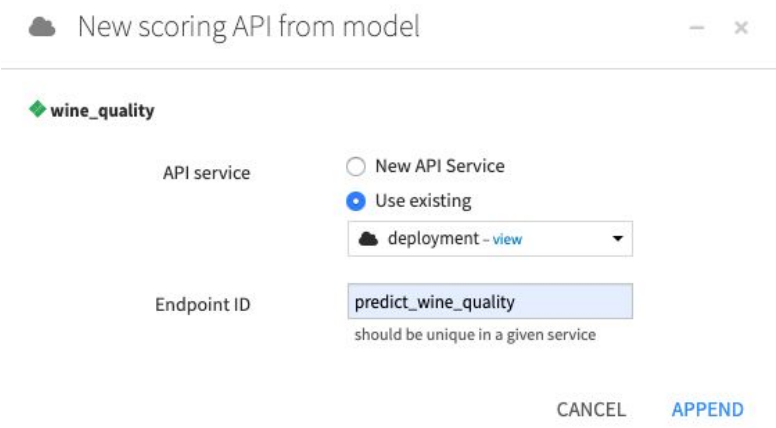
They need to monitor performance in real time

LeaveNoOneOut

Story 2: Deploy machine learning model to production

DSS Production features – from ML model to:

- automatic retraining
- batch scoring
- designing a REST API
- full-fledged production deployment (Kubernetes)
- python and java runtime
- exports as .jar or .pmml files



LeaveNoOneOut

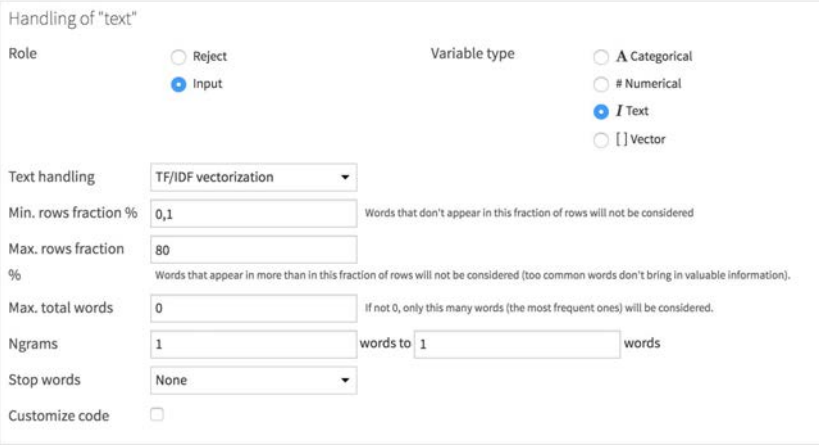
Story 2: deploy machine learning model to production

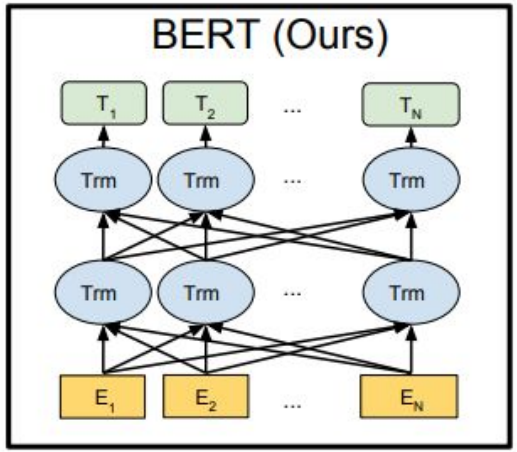
Under the hood, a scikit-learn model is converted to a java object

```
clf (DecisionTreeClassifier)
        ↓
    clf.tree_
        ↓
  zipped json file
        ↓
public class DecisionTreeModel
```

LeaveNoOneOut

Story 3: Packaging reusable Code





Data Analyst

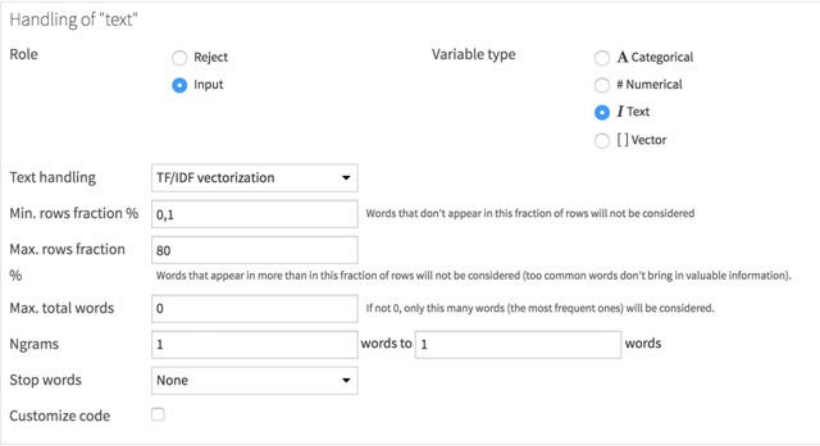
Data Scientist has a cool custom code

Data Analyst wants to try and tune it

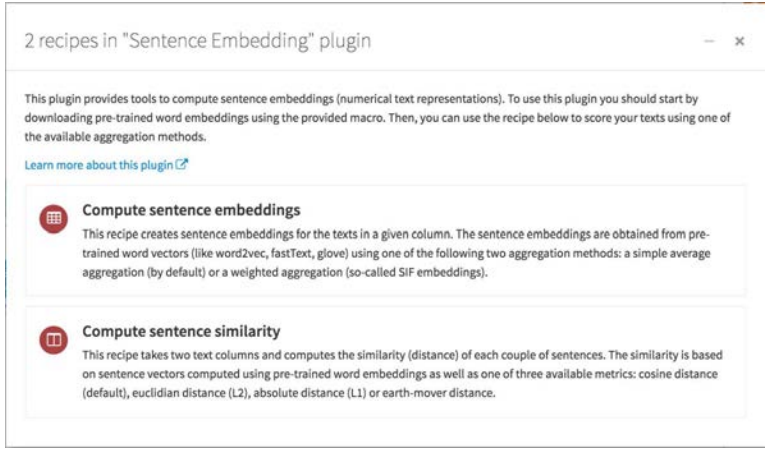
Data Scientist

LeaveNoOneOut

Story 3: Packaging reusable Code



Native Visual TF-IDF Processor

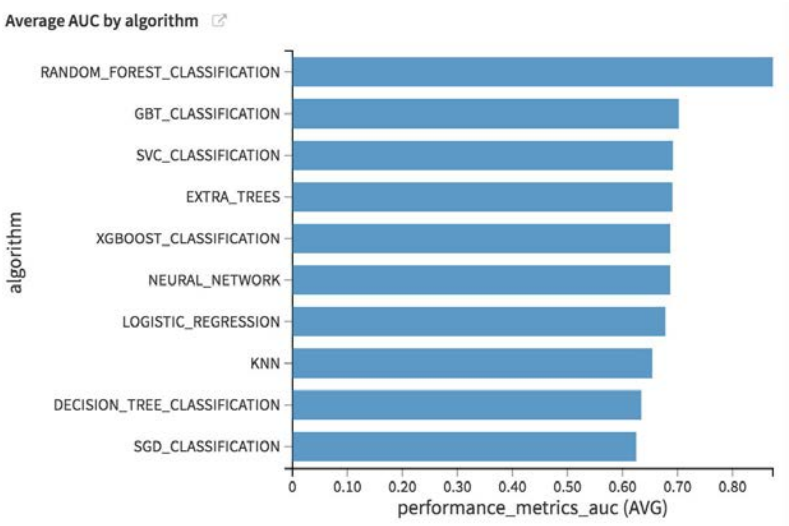


Sentence Embedding Plugins

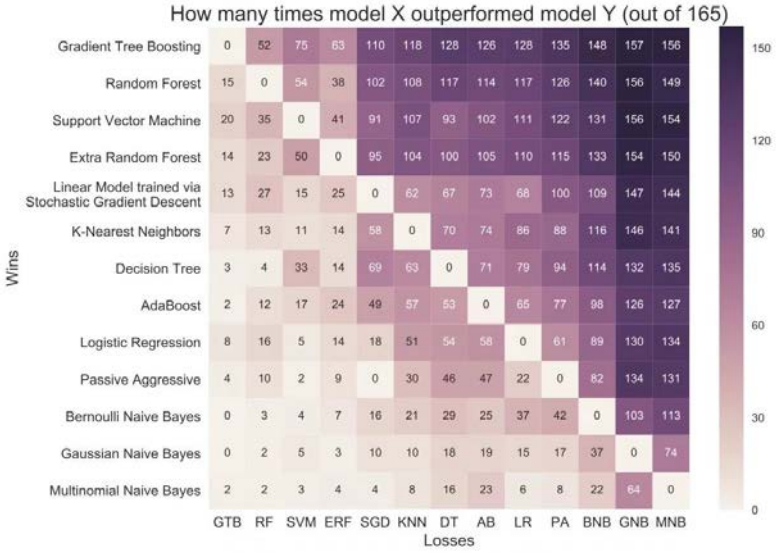
`sklearn_talk.dataiku`.CommonGroundBuilder

Project 1: Benchmarking Machine Learning Techniques

ML Models against PennML



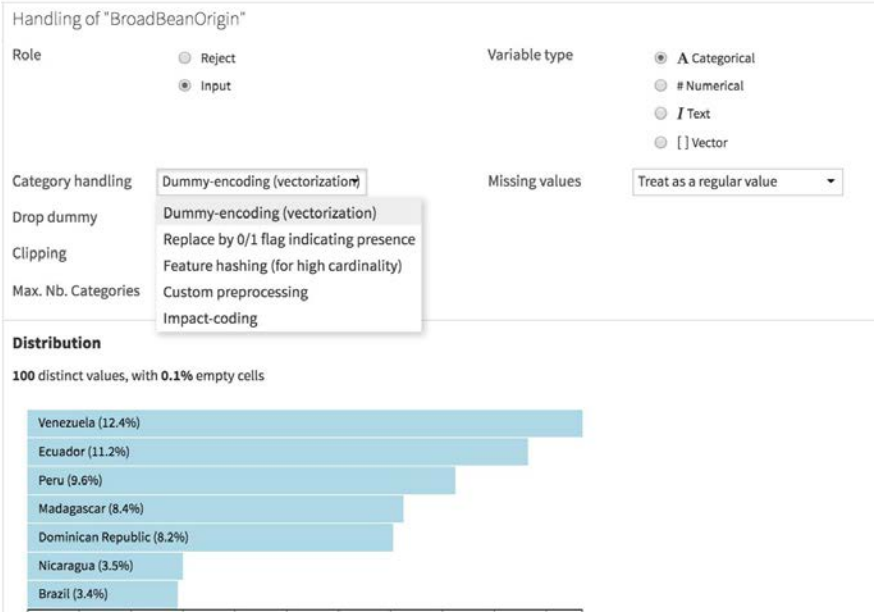
Over 165 PennML Classification Datasets



Data-driven Advice for Applying Machine Learning to Bioinformatics Problems, Olson et al.

Project 1: Benchmarking Machine Learning Techniques

Feature Representation Benchmark



dirty_cat

fastText

Project 1: Benchmarking Machine Learning Techniques

Imbalance Learning Benchmark

scikit-learn-contrib / **imbalanced-learn**

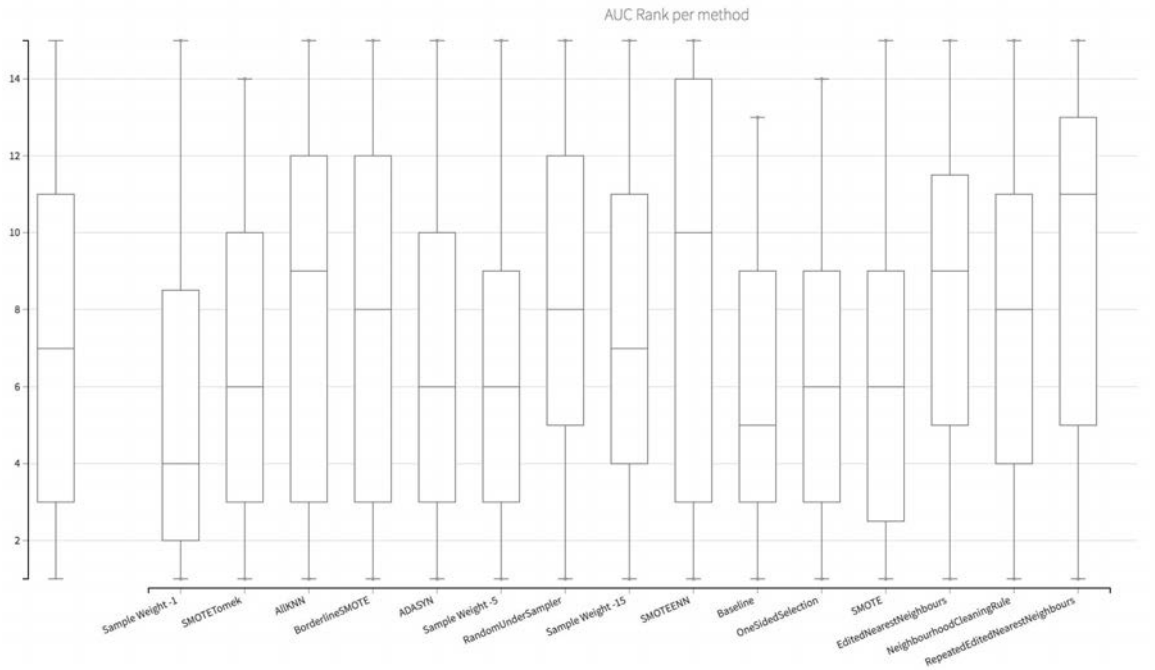
15 strategies from imbalanced-learn

32 datasets from OpenML

Project 1: Benchmarking Machine Learning Techniques

Imbalance Learning Benchmark

Work In Progess



Distribution of Strategies Ranking Across Datasets

LeaveNoOneOut

Project 2: Adding Sample Weights *Everywhere*

- Feature request: enable sample weights for (supervised) ML training and scoring
- Roadblock: sample weights are not supported *everywhere* in scikit-learn

<u>Solution 1:</u>

```
if sample_weights is not None:
    import statsmodels
    ...
```

<u>Solution 2:</u>

[MRG+1] Add sample weights support to kernel density estimation (fix #4394) #10803

Merged jnothman merged 19 commits into scikit-learn:master from samronsin:sample-weights-in-KDE on 26 Jun 2018

Some Ongoing Projects

Research at Dataiku

Distributed Hyperparameter search with Dask and Joblib

Automatic Feature Generation

Drift Detection

ML Interpretability

Reinforcement Learning (Beyond Video Games)

Active Learning for Smarter Annotations