Jupyter widgets for human-in-the-loop data science

@pascalbugnion
pbugnion
https://pascalbugnion.net

Pascal Bugnion
Strata London 2018
ASI Data Science is a London-based data science consultancy
About me

— Committer for Jupyter widgets
— Main author of jupyter-gmaps, a library for visualizing geographical data in Jupyter notebooks
— Author of Scala for data science (Packt Publishing)
Jupyter widgets for human-in-the-loop data science
Developing machine learning software is hard
Developing machine learning software is hard

— almost always stochastic
— often black box
Developing machine learning software is hard

— robustness
— overfitting
— overreliance on certain features or groups of features
Traditional software development workflows are inadequate.
Human intuition

It should be easy for humans to explore the model.
Human intuition

Computers think in terms of bytes and instructions, and humans think in terms of concepts and images.
Human intuition

We need a framework to rapidly create UIs that allow the human to think at a higher level of abstraction.

The UI should not be a black box.
Jupyter widgets
Simple spectral analysis

An illustration of the Discrete Fourier Transform

\[ X_k = \sum_{n=0}^{N-1} x_n \exp \left( -\frac{2\pi i}{N} k n \right) \quad k = 0, \ldots, N-1 \]

```python
In [2]: from scipy.io import wavfile
t = wavfile.read('test_mono.wav')
```

And we can easily view its spectral structure using matplotlib's `specgram` routine:

```python
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize=(16,5))
    
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```
Jupyter widgets allow building user interfaces entirely in Python, directly in Jupyter notebooks.
Build user interfaces in Python directly in Jupyter notebooks
Examples
Jupyter widgets

- Jupyter widgets are written entirely in Python
- They are written in the environment the data scientist is currently working in
- Widgets have access to the entire state of the notebook
Core: ipywidgets

```python
labeller = Labeller(transactions, CATEGORIES)
labeller.render()
```

<table>
<thead>
<tr>
<th>Memo</th>
<th>Correct</th>
<th>Predicted</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEW SOUTHERN RAILW LND SW1V 5973</td>
<td>✓</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>SARDO COFFEE SHOP LONDON W1T</td>
<td>✓</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>SICILIANA LONDON</td>
<td>✓</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>TASKRABBIT 2PHQ8 T 02086109298</td>
<td>✓</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>RELISH</td>
<td>✓</td>
<td>other</td>
<td></td>
</tr>
<tr>
<td>INT'L 0092766987 Amazon UK Retail A...</td>
<td>✓</td>
<td>travel</td>
<td></td>
</tr>
<tr>
<td>CASH RB SCOT OCT26 TESCO MORE ...</td>
<td>✓</td>
<td>food</td>
<td></td>
</tr>
<tr>
<td>TESCO STORES 5244 KILBURN</td>
<td>✓</td>
<td>food</td>
<td></td>
</tr>
<tr>
<td>TESCO STORES 3479 ST JOHNS WOOD</td>
<td>✓</td>
<td>food</td>
<td></td>
</tr>
<tr>
<td>TESCO STORES 2722 EUSTON</td>
<td>✓</td>
<td>food</td>
<td></td>
</tr>
<tr>
<td>retrain</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Linear Regression

Regression Line: 6.81 + 4.86x
ipyleaflet and gmaps
import numpy as np
import pandas as pd
import qgrid
randn = np.random.randn

df_types = pd.DataFrame({
                            index=list(range(9)), dtype='datetime64[ns]'),
    'B': pd.Series(randn(9), index=list(range(9)), dtype='float32'),
    'C': pd.Categorical(['washington', 'adams', 'madison', 'lincoln', 'jefferson', 'hamilton', 'roosevelt',
                         'foo', 'bar', 'bippity', 'boppy', 'foo', 'foo', 'bar', 'zoo']),
    'D': ["foo", "bar", "buzz", "bippity", "boppy", "foo", "foo", "bar", "zoo"]})
df_types['E'] = df_types['D'] == 'foo'
qgrid_widget = qgrid.QgridWidget(df=df_types, show_toolbar=True)
qgrid_widget
```python
In [1]: import ipyvolume

In [2]: ipyvolume.examples.ball(rmax=3, rmin=2.5, shape=64, lighting=True)
```
Learning about widgets

- **mlviz**: visualising machine learning algorithms with Jupyter widgets and bqplot.
- Jupyter widgets tutorial
- Coding a simple widget from scratch: video and code
- Jupyter widgets documentation
Libraries used in this talk

- ipywidgets
- bqplot
- gmaps
- lens
- superintendent
Use widgets to reduce friction at the human computer interface
Acknowledgements

— Jupyter widgets developers: Jason Grout, Sylvain Corlay, Maarten Breddels, Matt Craig, Vidar Tonaas Fauske
— ASI Data Science
— Chakri Cherukuri (ChakriCherukuri)
— Victor Zabalza (zblz) and Scott Stevenson (srstevenson)
— Jan Freyberg (janfreyberg)
— SherlockML