

Interactive

September 4, 2019

Data Science Workflow < | > Hall of Fame

1 Exploring Subsets Interactively

In this notebook you can select criteria for a subset to inspect and compare to the entire dataset.

Instructions:

1. Run Cell 1, wait for "DONE IN <time>" message before continuing.
2. Run Cell 2, fill out criteria before continuing.
3. Run Cell 3, view report.
4. Repeat steps 2 and 3 with new criteria, if desired.

Prerequisites:

- four final CSV file local in ./data_final
 - all aggregations created by aggregate.py local in ./analysis_data
 - ipywidgets
 - pip install ipywidgets
 - nodejs
 - conda install nodejs
 - npm
 - pip install npm
 - labextension
 - for jupyterlab: jupyter labextension install @jupyter-widgets/jupyterlab-manager
 - for jupyter notebook: notebook extension (jupyter nbextension enable --py widgetsnbextension)
-

1.1 Run Cell 1

This takes about 10 minutes. Wait for "DONE IN <time>" message before continuing!

```
[1]: import interactive
import load_data
import datetime
from IPython.core.display import HTML
```

```
# 10 minutes to load data
start = datetime.datetime.now()
data_frames = interactive.data()
end = datetime.datetime.now()
print('\n'+ '-'*80+'\n'+ 'DONE IN {0}'.format(end - start))
```

```
Notebooks loaded in 0:00:24.846901
Repos loaded in 0:00:04.234393
Owners loaded in 0:00:00.613096
Notebook imports loaded in 0:00:40.402695
Errors loaded in 0:00:02.953865
Cell stats loaded in 0:00:01.658813
Cell order loaded in 0:00:26.470042
Outputs loaded in 0:00:01.949563
Statuses loaded in 0:00:01.080600
Cell stats loaded in 0:00:01.935130
Collaboration statuses loaded in 0:00:00.028222
Special functions loaded in 0:00:11.965535
Framework uses loaded in 0:00:08.903738
Educational status loaded in 0:00:00.194457
```

```
-----
DONE IN 0:09:12.868796
```

```
[4]: # query = interactive.interactive(data_frames)
```

1.2 Run Cell 3

This takes about 1 minute. View report! Repeat Cells 2 and 3 with different criteria.

```
[3]: data_frames_sub = interactive.subset(data_frames, query)
print('\n'+ '-'*73+'\n')
interactive.report_comparisons(data_frames_sub, data_frames)
```

```
Subsetting to Python notebooks pushed between 2011-10-24 and 2019-07-14.
Notebooks considered use at least one machine learning framework and at least
one visualization package.
962,959 (24.79%) notebooks fit your criteria.
```

1.2.1 Summary Statistics

	num_cells	forks_count	open_issues_count	stargazers_count	\
mean	39.2	5.19	0.23	9.36	
median	28.0	0.00	0.00	0.00	

min	0.0	0.00	0.00	0.00
max	1641.0	8589.00	559.00	18885.00

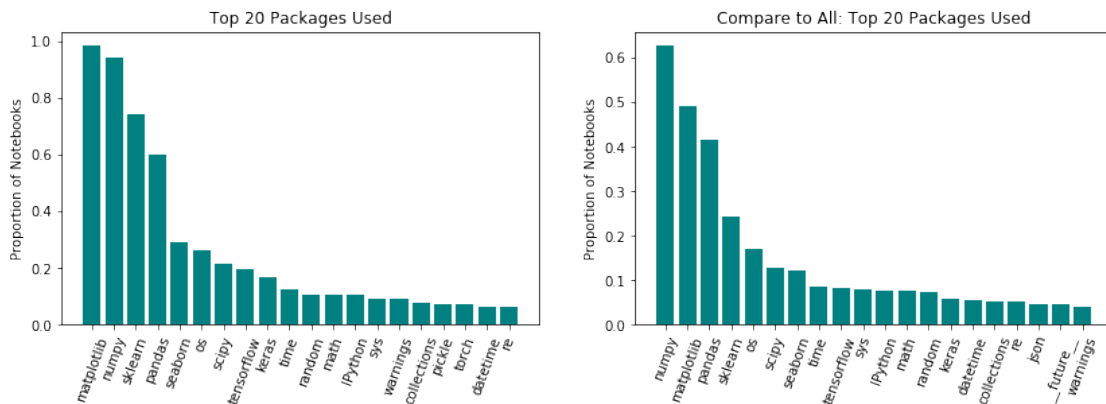
	subscribers_count	watchers_count	lines_of_code	num_words
mean	1.67	9.36	222.86	529.73
median	1.00	0.00	157.00	98.00
min	0.00	0.00	1.00	0.00
max	2446.00	18885.00	40759.00	55037.00

Compare to all:

	num_cells	forks_count	open_issues_count	stargazers_count	\
mean	28.76	5.56	0.49	9.88	
median	19.00	0.00	0.00	0.00	
min	0.00	0.00	0.00	0.00	
max	1641.00	17634.00	2003.00	22831.00	

	subscribers_count	watchers_count	lines_of_code	num_words
mean	2.13	9.88	147.26	405.18
median	1.00	0.00	88.00	60.00
min	0.00	0.00	1.00	0.00
max	2446.00	22831.00	462118.00	200404.00

1.2.2 Package Use

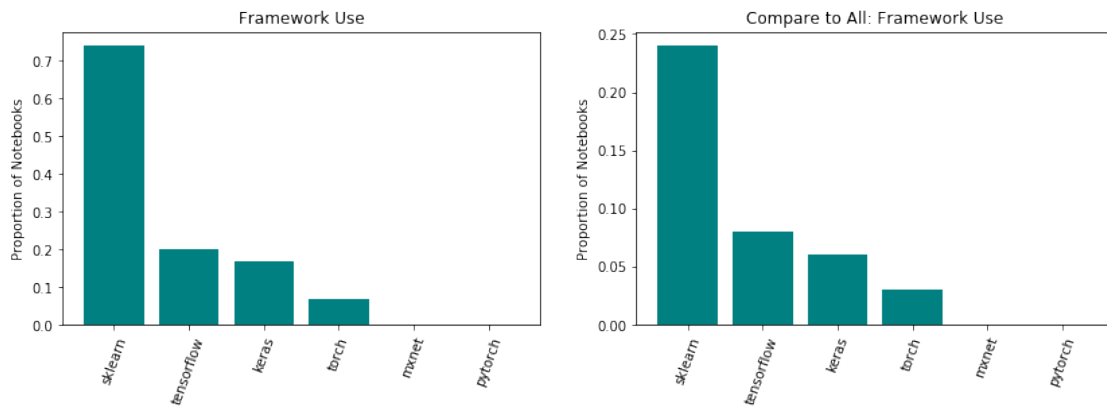


1.2.3 Framework Use

100.0% of these notebooks use at least one framework.

Compare to all:

35.48% of all notebooks use at least one framework.

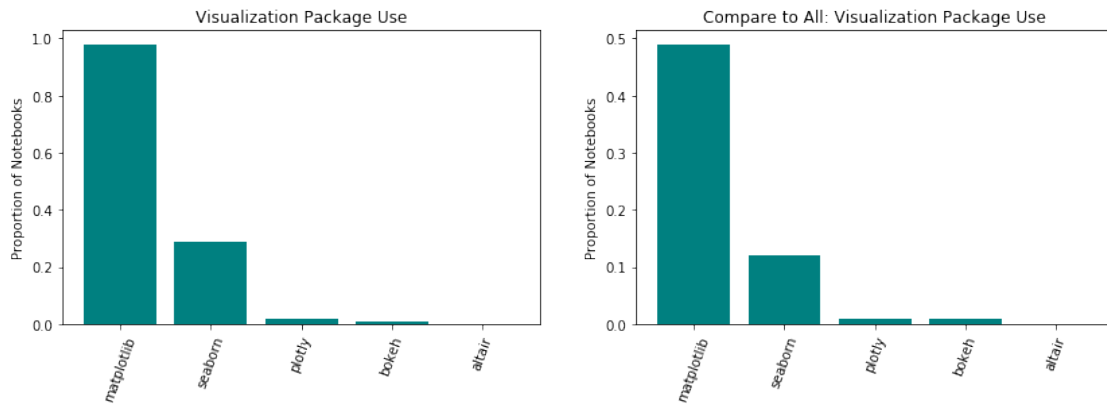


1.2.4 Visualization Package Use

100.0% of these notebooks use at least one visualization package.

Compare to all:

50.71% of all notebooks use at least one visualization package.

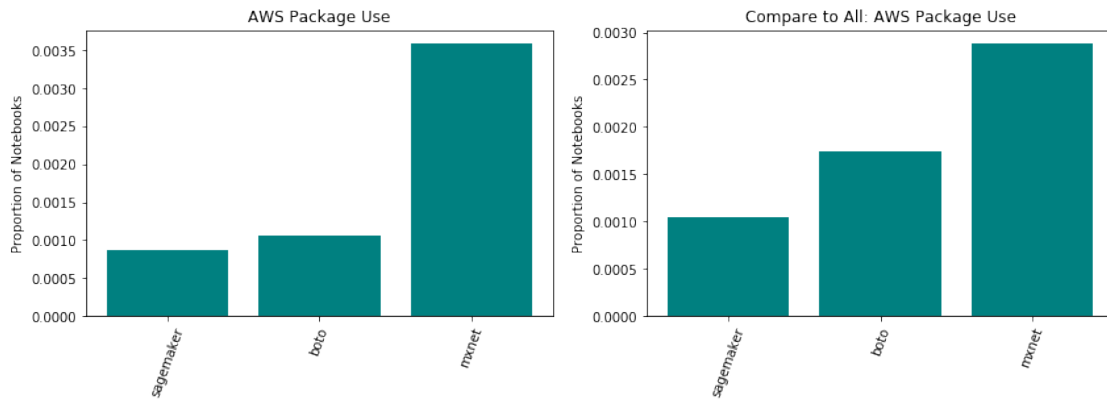


1.2.5 AWS Package Use

0.488% of these notebooks import at least one of sagemaker, boto, or mxnet.

Compare to all:

0.493% of all notebooks import at least one of sagemaker, boto, or mxnet.



1.2.6 Number of Errors per Notebook

```
mean      0.21
median    0.00
min       0.00
max      9104.00
Name: num_errors, dtype: float64
```

Compare to all:

```
mean      0.22
median    0.00
min       0.00
max      9104.00
Name: num_errors, dtype: float64
```

1.2.7 Ratio of Markdown to Code

```
mean      3.27
median    0.65
min       0.00
max      964.22
Name: ratio_wl, dtype: float64
```

Compare to all:

```
mean          6.45
median        0.69
min           0.00
max          18029.00
Name: ratio_wl, dtype: float64
```

1.2.8 Execution Order

67.88% of these notebooks have cells run in order.

91.13% of these notebooks have at least one output, 64.85% of which are run in order.

86.58% of these notebooks were able to be parsed with Python AST.

Of these, 14.38% had a function used before it was defined, 3.45% had a package used before it was imported, and 6.21% used a variable before it was defined.

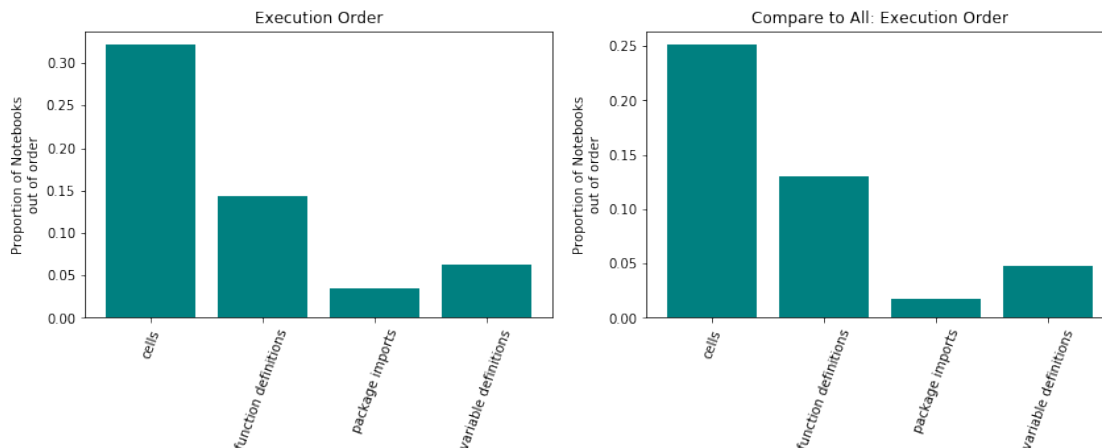
Compare to all:

74.92% of all notebooks have cells run in order.

84.4% of all notebooks have at least one output, 70.58% of which are run in order.

86.41% of all notebooks were able to be parsed with Python AST.

Of these, 13.0% had a function used before it was defined, 1.8% had a package used before it was imported, and 4.76% used a variable before it was defined.

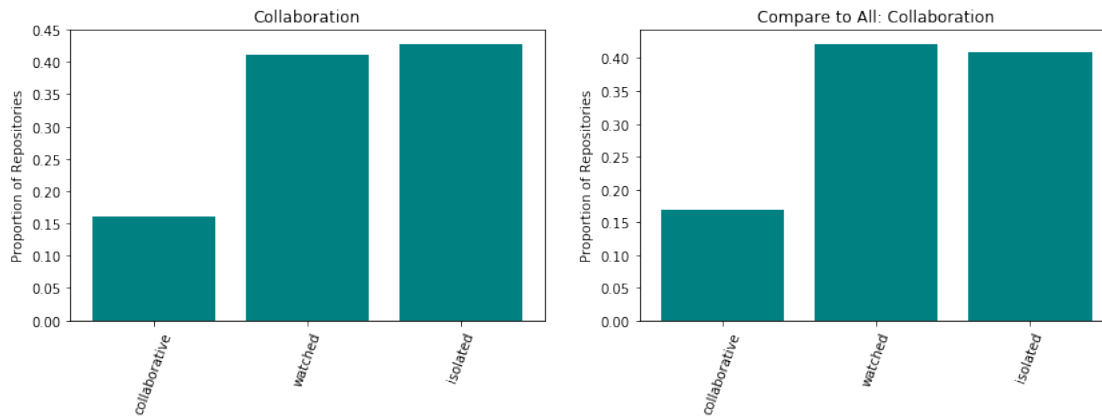


1.2.9 Collaboration

16.14% of these repositories are collaborative, containing 20.43% of these notebooks.

Compare to all:

16.85% of all repositories are collaborative, containing 22.76% of all notebooks.



1.2.10 Educational Status

24.42% of these repos are educational, holding 29.39% of these notebooks

Compare to all:

23.67% of all repos are educational, holding 29.21% of all notebooks

Data Science Workflow < | > Hall of Fame