Odyssey

Scikit-learn analysis meeting notes

## 

## January 25th 2018

Todo:

* Compare usage of skearn, pandas, matplotib, numpy between ipynb and python files.
* Compare different modules, functions, classes between ipynb and python files.(sklearn)
* Find sklearn.pipeline.Pipeline sklearn.pipeline.make\_pipeline usage. Give some examples. Make sure not to get false positives.
* Find from where the func is imported using a parser
* Functions using in ipynb and py

Previous analysis done:

<https://github.com/alan97/analysis-of-sklearn>

Setup.py is broken -> create fork, make pull-request with fix.

Fix the assert sts.

Run the codes for all the modules of sklearn (in odyssey)

Which libraries are used together

Hyperparameters used for function(eg. RF)

Pipeline:

See if the objects used for pipeline are scikit learn object

Other libraries which have “pipeline” should be ignored

**Updates:**

1. The assert statements has been fixed.
2. Codes done for classes, functions and submodules of Sklearn- repo which imported and count.
3. Libraries used with sklearn analysis-

otherModuleList = ['sqlalchemy','matplotlib','pandas', 'numpy', 'theano', 'nltk', 'tensorflow', 'pylearn2', 'theanets', 'hebel', 'caffe', 'pybrain', 'brainstorm', 'liblinear', 'libsvm', 'statsmodels', 'scipy']

All the import statement in the repository were extracted, where the file contained sklearn imports.

1. For all the functions in sklearn, the function calls were analysed to understand the hyperparameters used for each.

Thursday 2/15/2018

* Why do pipeline counts mismatch?
* Reproduce counts of classes in sklearn used
* Count how often other libraries are used with sklearn

Targets for 1st March:

1. Run “ALL” the queries in the new data.
2. Plot the classes used in py vs ipynb.
3. Using the parso for the hyperparameters.

### Pipeline usage- > usages without make\_pipeline

\*\*\* Do a very GOOD documentation , send pull request to ALAN for verification!!

Note all other libraries with pipeline

**Updates: Meeting- 1st March**

1. All the data has been run and all contents saved locally.
2. Data being parsed by parso.
3. Hyperparameter search using parso/ast.
4. Pipeline- can be captured. As the list contains the import of the library and function usage.

## March 8th

Discuss ast vs parso / py2 compatibility

One\_page\_v2 last part- instantiation with jedi !

Scatter plot !

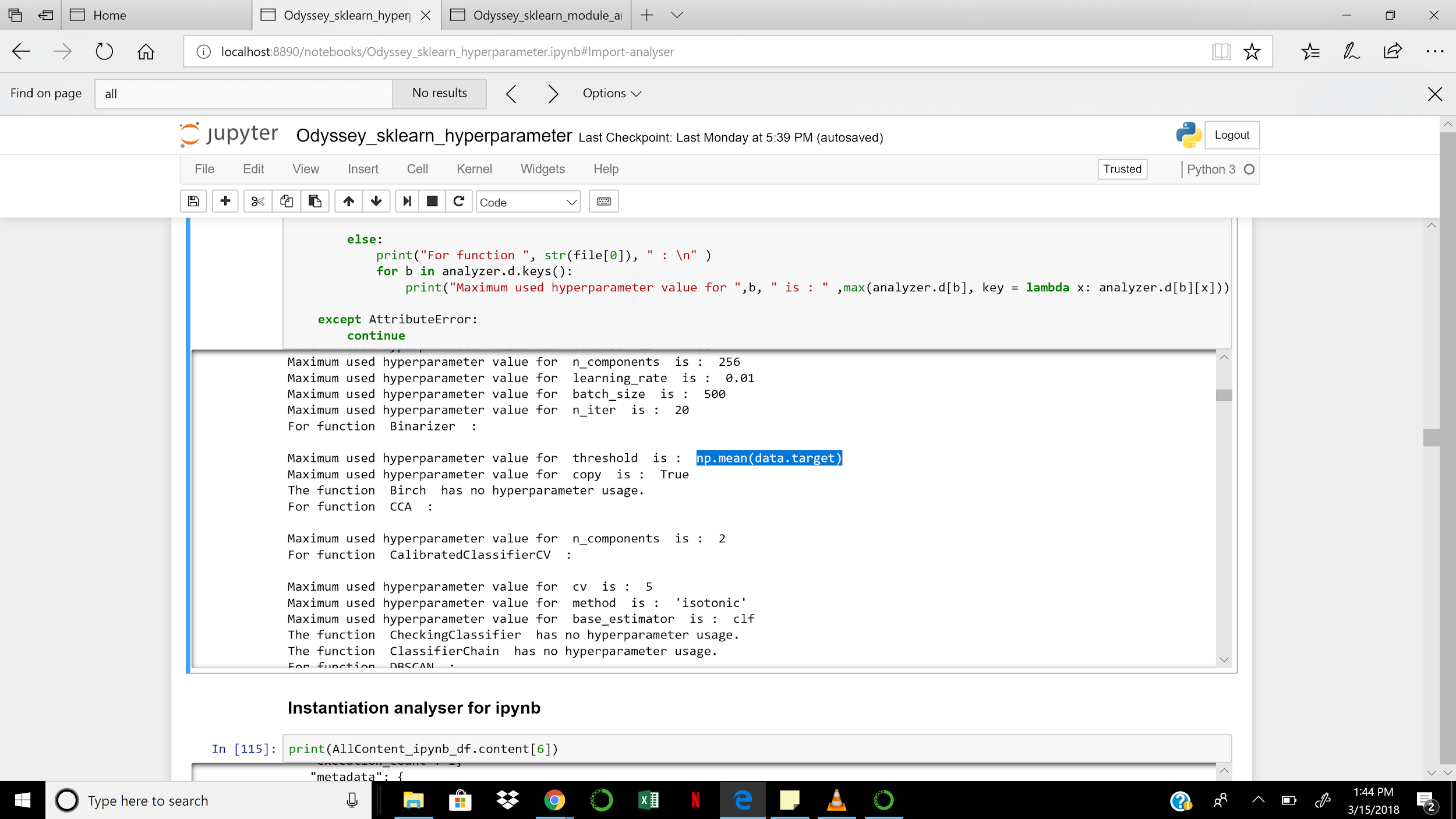
Import Analyser

* Functions like getInstantiation, getContextAll etc. in one-page\_v2 are deprecated from the instantiationanalyser !
* The function work with the query sent using bigquery. Hence, not an efficient way to do it.

Done

* Instantiation analyser used for all hyperparameters for all functions in content\_py\_unique and \*\*\* content\_ipynb.
* Scatter plot for number of imports of sklearn modules in ipynb and py files.

March 15th Meeting :



To Do :

## (Done )Count along with the max

## (Done) Save all the hyperparameter usage

## Jedi.names

Script : {completions(), goto\_definitions(), goto\_assignments(), usages(), call\_signatures()}

Using nb convert - convert the json files of ipynb to py files

“Pipeline” usage - integration of ImportAnlayser and Instantiation Analyser

“GridSearchCV” - models + parameters - > using parso find the model declared variables; also the hyperparameters declared as variables and passed to the GridSearchCV.

**April 5th Updates:**

* The function usages of all Sklearn instances extracted in both ipynb files and py\_unique files.
* Instantiation of the object/ not the usages and other calls !
* Usage comparison of the objects of various function (The actual pipeline use, vs the one calculated earlier)
* Instantiation Analyser on both Pipeline and MakePipeline. // double check if jedi captures make\_pipeline)
* Put data on Figshare.

Download another file using pipeline and test with instantiation analyser and jedi. (check for the tensorflow file)

**Pending:**

* **Inferential analysis on the hyperparameter usages !**
* **Understand function usages of Pipeline**
* **Scikit learn documentation of calibration\_loss**

**-> In brier score add what is cal loss and refinement loss.**

Generate data on the fly which gives infinity values during predict proba.

* Change editor to detect the indentation and other errors. Pep8
* Indexes of the files which have BRSM used.
* RFC with pipeline -> which file ? usage?
* Complete a report. For everything