

## Table of Contents

<b>Machine-Learning-Specific Refactoring Types.....</b>	<b>2</b>
Data Handling Optimization .....	2
Model Initialization Refinement .....	2
Resource Allocation Optimization .....	3
Data Type Clarification .....	3
Data Presentation Enhancement .....	4
Data Path Management .....	4
Mathematical Operation Refactoring.....	4
<b>General Refactoring Types .....</b>	<b>5</b>
Code Cleanup.....	5
Code Simplification .....	5
Import/Export Optimization.....	6
Logging Enhancement .....	6
Requiremesnt/Configuration Update .....	7
Condition Simplification.....	7
File/Dependency Path Refinement .....	8

# Machine-Learning-Specific Refactoring Types

## Data Handling Optimization

### Description:

- Optimizing data reading/writing from/to the dataset.

### Example:

- <https://github.com/chemprop/chemprop/commit/ae8ae8af6ad4e92c4edae62699fb655a77a3b0b7>



```
@@ -97,26 +97,55 @@ def get_smiles(path: str) -> List[str]:
97 97         :return: A list of smiles strings.
98 98         """
99 99         with open(path) as f:
100 -             f.readline() # Skip header
101 -             smiles = [line.strip().split(',')[0] for line in f]
100 +             reader = csv.reader(f)
101 +             next(reader) # Skip header
102 +             smiles = [line[0] for line in reader]
102 103
103 104         return smiles
```

⇒ The refactoring is accomplished by replacing manual line reading and header skipping with the “csv.reader” object, specifically designed for CSV file reading.


## Model Initialization Refinement

### Description:

- Modify the hyper-parameter initializations in the models.

### Example:

- <https://github.com/scikit-learn/scikit-learn/commit/14ecaa19c66d2af94e110268410e903fc8ffc6dd>



```
@@ -182,10 +182,13 @@ def test_base_hmm_attributes(self):
182 182
183 183
184 184     def train_hmm_and_keep_track_of_log_likelihood(hmm, obs, n_iter=1, **kwargs):
185 -         hmm.fit(obs, n_iter=1, **kwargs)
185 +         hmm.n_iter = 1
186 +         hmm.fit(obs)
186 187         loglikelihoods = []
187 188         for n in xrange(n_iter):
188 -             hmm.fit(obs, n_iter=1, init_params='', **kwargs)
189 +             hmm.n_iter = 1
190 +             hmm.init_params = ''
191 +             hmm.fit(obs)
```

⇒ The refactoring involves breaking down the model function call into individual steps, explicitly setting the parameters “n\_iter” and “init\_params” before calling the “fit” method to match the library API.

## Resource Allocation Optimization

### Description:

- Adjusting hardware usage parameters.

### Example:

- <https://github.com/mars-project/mars/commit/b2316ea69686dd5246c46cd8a47f6f67d8f42faa>

```
▼ 5 mars/deploy/kubernetes/tests/test_kubernetes.py
...
160 160 @@ -160,7 +160,8 @@ def _start_kube_cluster(use_test_docker_file=True, **kwargs):
161 161     @pytest.mark.skipif(not kube_available, reason='Cannot run without kubernetes')
162 161     def test_run_in_kubernetes(use_test_docker_file):
163 162         with _start_kube_cluster(
163 163             - worker_mem='1G', worker_cache_mem='128m',
164 164             + supervisor_cpu=0.5, supervisor_mem='1G',
165             + worker_cpu=0.5, worker_mem='1G', worker_cache_mem='64m',
166             extra_labels={'mars-test/group': 'test-label-name'},
167         ):
```

- ⇒ The refactoring in this code involves adding more specific parameters for CPU allocation “supervisor\_cpu” and “worker\_cpu” and adjusting memory allocations “worker\_cache\_mem” while maintaining the overall functionality of the “\_start\_kube\_cluster” function call.

## Data Type Clarification

### Description:

- Converting datatypes from one to another (e.g., from NumPy to DataFrame).

### Example:

- <https://github.com/q-optimize/c3/commit/2ea2f0b9639ea572458563f3b53d06505d4a4e2f>

```
31 38         self.pmap = pmap
32 38         - self.optim_status = {}
33 38         - self.gradients = {}
39 39         + self.optim_status: Dict[str, Any] = dict()
40 39         + self.gradients: Dict[str, np.ndarray] = {}
34 41         self.current_best_goal = 9876543210.123456789
```

- ⇒ This change introduces type hinting by explicitly specifying the type of “self.optim\_status” as a dictionary “Dict[str, Any]” and initializing it with an empty dictionary using “dict()”. This can improve code readability and help catch type-related errors early during development.

## Data Presentation Enhancement

### Description:

- Refactoring data visualization code.

### Example:

- <https://github.com/biolab/orange3/commit/c2d3a8a30a5e68a13512bdc5ad85a5ba0b2a0053>

```
65 - box = gui.widgetBox(self.controlArea, "Selection")
61 + box = gui.widgetBox(self.controlArea, "Select")
```

⇒ renaming “Selection” to “Select” in the widget box title aimed at improving the clarity and consistency of the plot.

## Data Path Management

### Description:

- Managing dataset/plots/models storage paths.

### Example:

- <https://github.com/automl/auto-sklearn/commit/2388087e5b347ae7d6425afaf878a8f279bd663e>

```
50 49 def main():
51 - datasets = 'resources/datasets.csv'
50 + datasets = 'datasets.csv'
```

⇒ moving the CSV dataset file into a “resources” directory can make the code more organized and easier to manage, which doesn’t change external functionalities.

## Mathematical Operation Refactoring

### Description:

- Changing mathematical calculation to a simpler form.

### Example:

- <https://github.com/pyro-ppl/pyro/commit/4f5940971cb0c7c42f81450ce70b6244551ace02>

```
pyro/distributions/hmm.py
125 125 result = _sequential_logmatmulexp(result)
126 126
127 127 # Combine initial factor.
128 - result = _logmatmulexp(self.initial_logits.unsqueeze(-2), result).squeeze(-2)
128 + result = self.initial_logits + result.logsumexp(-1)
```



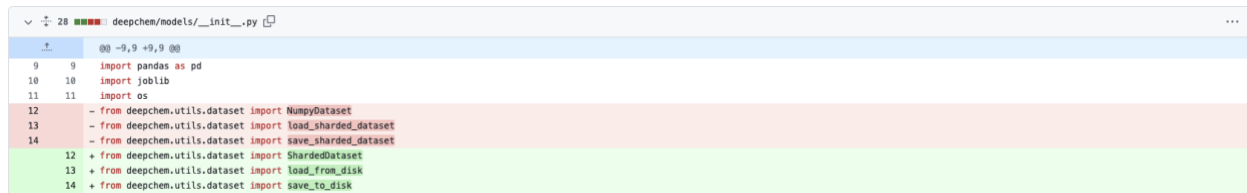
## Import/Export Optimization

### Description:

- Removing or relocating unused dependencies.

### Example:

- <https://github.com/deepchem/deepchem/commit/25f19f49668cf0ff28a87e8aa4719f5d429d299b>



```
28 deepchem/models/__init__.py
9 9 import pandas as pd
10 10 import joblib
11 11 import os
12 12 - from deepchem.utils.dataset import NumpyDataset
13 13 - from deepchem.utils.dataset import load_sharded_dataset
14 14 - from deepchem.utils.dataset import save_sharded_dataset
12 12 + from deepchem.utils.dataset import ShardedDataset
13 13 + from deepchem.utils.dataset import load_from_disk
14 14 + from deepchem.utils.dataset import save_to_disk
```

⇒ Changing multiple imports with a similar library in data-load/featurization.

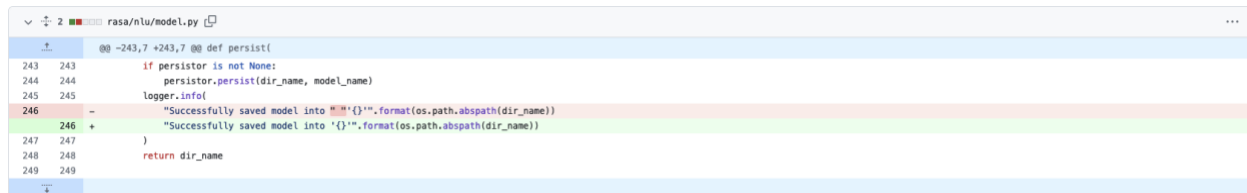
## Logging Enhancement

### Description:

- Improving log messages for user understanding or modify them for readability/consistency.

### Example:

- <https://github.com/RasaHQ/rasa/commit/61ba44a1be9e674566ab20930f217e8fdd363ebb>



```
2 2 rasa/nlu/model.py
243 243 if persistor is not None:
244 244 persistor.persist(dir_name, model_name)
245 245 logger.info(
246 246 - "Successfully saved model into {}".format(os.path.abspath(dir_name))
246 246 + "Successfully saved model into '{}'.format(os.path.abspath(dir_name))
247 247 )
248 248 return dir_name
249 249
```

⇒ improves the readability and consistency of the code by removing redundant string concatenation and using a single format specifier for the placeholder.

## Requiresnt/Configuration Update

### Description:

- Updating program requirements or configurations.

### Example:

- <https://github.com/hi-primus/optimus/commit/436f7f1923ba6fd8cc410f16ab93461ccfc0e37a>

```
requirements.txt
11 11 @ @ ~11.7 +11.0 @ nose=1.3.7
12 12 ipython=6.5.0
13 13 seaborn=0.8.1
14 14 setuptools=40.1.0
15 14 - quinc=0.2.1
16 15 deprecated=1.2.0
17 16 pyarrow=0.8.*
18 16 tabulate=0.8.2
19 19 @ @ ~21.5 +20.4 @ keras=2.1.5
20 20 pillow=4.1.1,4.2
21 21 pygments=2.2.0
22 22 six=1.10.0
23 23 - h5py=2.7.0
24 24 - timber
25 25 + h5py=2.7.0
```

⇒ Remove unused dependencies from the requirements.

## Condition Simplification

### Description:

- Simplifying complex conditional statements.

### Example:

- <https://github.com/ray-project/ray/commit/8f59546ef2fba5af2666c27e5480f6819389fedf>

```
104 - if row_metadata is not None:
105 -     self._row_metadata = row_metadata.copy()
106 - if col_metadata is not None:
107 -     self._col_metadata = col_metadata.copy()
108 105 assert self._block_partitions.ndim == 2, \
109 106     "Block Partitions must be 2D."
110 107 else:
111 108     if row_partitions is not None:
112 109         axis = 0
113 110         partitions = row_partitions
114 - if row_metadata is not None:
115 -     self._row_metadata = row_metadata.copy()
116 111 elif col_partitions is not None:
117 112     axis = 1
118 113     partitions = col_partitions
119 - if col_metadata is not None:
120 -     self._col_metadata = col_metadata.copy()
121 114 self._block_partitions = \
122 115     _create_block_partitions(partitions, axis=axis,
123 116                             length=len(columns))
124 117
125 118
119 + if row_metadata is not None:
120 +     self._row_metadata = row_metadata.copy()
121 + if col_metadata is not None:
122 +     self._col_metadata = col_metadata.copy()
123 +
```

⇒ The code combines several nested conditions into two straightforward conditions.

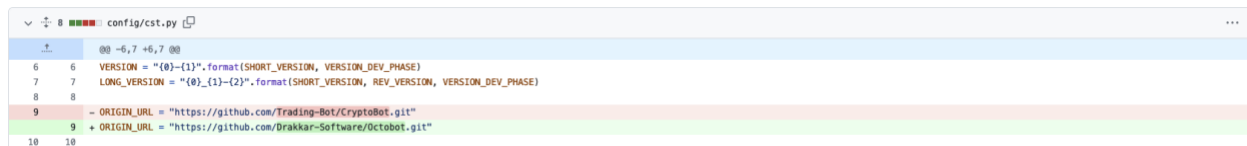
## File/Dependency Path Refinement

### Description:

- Optimizing/updating dependency/file path.

### Example:

- <https://github.com/Drakkar-Software/OctoBot/commit/3ce0df4d77976a0cd78ada75803f7fa5b8b9f522>



```
config/cst.py
6 6 VERSION = "{0}-{1}".format(SHORT_VERSION, VERSION_DEV_PHASE)
7 7 LONG_VERSION = "{0}_{1}-{2}".format(SHORT_VERSION, REV_VERSION, VERSION_DEV_PHASE)
8 8
9 - ORIGIN_URL = "https://github.com/Trading-Bot/CryptoBot.git"
9 + ORIGIN_URL = "https://github.com/Drakkar-Software/Octobot.git"
10 10
```

⇒ Change the directory of Cryptobot with Octobot