[Model Memory Optimization](https://inside-docupedia.bosch.com/confluence/display/BSE/Model+Memory+Optimization) in class

**slots Technique:**

Python uses dictionaries to store instance attributes of instances of a class. This allows you to dynamically add more attributes to instances at runtime but also create a memory overhead. In Python is a special mechanism that is used to reduce memory of the objects. slots is a static type of method in this no dynamic dictionary are required for allocating attribute. The \_\_slots\_\_ optimizes the memory if the class has many objects.

When we are using slots, it will not allow to create new property in class. In ADE, when we are starting the model, we don’t need to create new properties. slots reduce in memory space as well as reduce time complexity.

Approx. 5% reduction, we are doing after applying slots. slots has inheritance property. Implementation of slots is straight forward. Implementation is done and tested on Original Code base.

**Experimentation: In-Memory Optimization**



**Experimentation: Time Optimization**



**Notes for ADE developers:**

Currently, the slots technique has been applied to the following classes:

* class Classifier(object) in classifiers.py
* class ModelConfig\_JSON(object) in iCompass\_ModelSettings.py
* class AnomalyScoringEngine(object) iCompass\_OnlineScoringEngine.py
* class RulesPortfolio(object) in iCompass\_RulesPortfolio.py
* class AnomalyDetectionEngine(object) in  in create\_model.py

When extending such classes (i.e., to add a new Rule), the slot definition should be updated.

* **Reference Link:**
* [**https://stackoverflow.com/questions/472000/usage-of-slots**](https://stackoverflow.com/questions/472000/usage-of-slots)
* [**https://www.pythontutorial.net/python-oop/python-\_\_slots\_\_/**](https://www.pythontutorial.net/python-oop/python-__slots__/)

**Compression Technique:**

We are currently saving model as pickle file. In these pickle file so many redundant keys are present. So, idea is can we somehow compressed these pickle file without losing any information? LZMA compression technique is lossless compression technique. LZMA compresses files using both statistical modeling and dictionary techniques. Statistical modeling allows for the analysis of entire blocks of text, whereas dictionary techniques compress small pieces simultaneously.

Python has in-built package for compressing pickle file using LZMA compression technique (compress\_pickle). I have directly changed in local code and run in local. I can see compressed pickle file which is less in size than simple pickle file. I ran model after changing the code for hours and results are following:

iCompassModel\_v1 (Compressed Pickle file) - 12 mb

iCompassModel\_v1 (Simple Pickle file) - 144 mb

I have also checked that after compression we are not losing any information. This is reducing the file size in disk.

**Test 1:** If model is already running then first compressing that pickle file and deleting pickle file.

**Test 2:** If model is not running then it will started from compressed pickle file.

**Test 3:** If monitor automatically crashed then it will take another compressed pickle file and run ADE code.

**Notes for ADE developers:**

Currently, LZMA compression technique is used to compress the pickle file. I have added compression technique in following python files:

* **create\_model.py**
* **main.py**
* **OnlineScoringEngine.py**

**deepdiff library is used to check pickle file and compressed pickle file lossless compression.**

**Reference Link:**

* <https://pypi.org/project/compress-pickle/>
* <https://www.youtube.com/watch?v=j2HSd3HCpDs>
* <https://www.youtube.com/watch?v=RV5aUr8sZD0>
* <https://en.wikipedia.org/wiki/Lempel%E2%80%93Ziv%E2%80%93Markov_chain_algorithm>
* <https://pypi.org/project/deepdiff/>