PyMLToolKit Documentation

Release latest

Feb 13, 2020

Contents

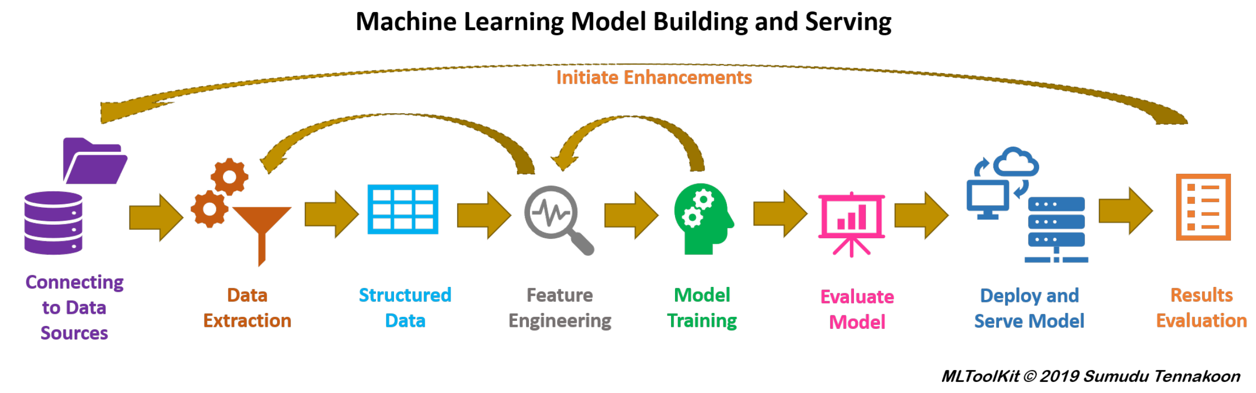
1 Cite as 3

2 Contents 5
2.1 Introduction 5
2.2 Install 6

3 Indices and Tables 9

MLToolKit (mltk) is a Python package providing a set of user-friendly functions to help building end-to-end machine learning models in data science research, teaching or production focused projects.

MLToolKit supports all stages of the machine learning application development process.



Project Home : https://mltoolkit.github.io/MLToolKit

CHAPTER 1

Cite as

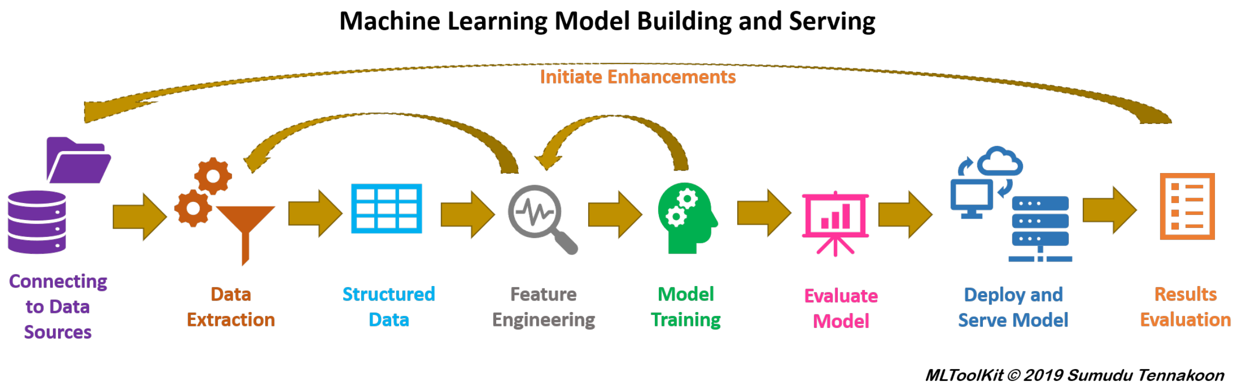
```
@misc{mltk2019,
  author =  "Sumudu Tennakoon",
  title = "MLToolKit(mltk): A Simplified Toolkit for Unifying End-To-End Machine␣
↪Learning Projects",
  year = 2019,
  publisher = "GitHub",
  howpublished = {\url{https://mltoolkit.github.io/mltk/}},
  version = "0.1.10",
  doi = "https://doi.org/10.5281/zenodo.3596163"
}
```

CHAPTER 2

Contents

2.1 Introduction

MLToolKit supports all stages of the machine learning application development process.



2.1.1 Functions

- Data Extraction (SQL, Flatfiles, Binary Files, Images, etc.)
- Exploratory Data Analysis (statistical summary, univariate analysis, visulize distributions, etc.)
- Feature Engineering (Supports numeric, text, date/time. Image data support will integrate in later releases of v0.1)
- Model Building (Currently supported for binary classification and regression only)
- Hyper Parameter Tuning [in development for v0.2]
- Cross Validation (will integrate in later releases of v0.1)

- Model Performance Analysis, Explain Predictions (LIME and SHAP) and Performance Comparison Between Models.
- JSON input script for executing model building and scoring tasks.
- Model Building UI [in development for v0.2]
- ML Model Building Project [in development for v0.2]
- Auto ML (automated machine learning) [in development for v0.2]
- Model Deploymet and Serving [included, will be imporved for v0.2]

2.1.2 Supported Machine Learning Algorithms/Packages

- RandomForestClassifier: scikit-learn
- LogisticRegression: statsmodels
- Deep Feed Forward Neural Network (DFF): tensorflow
- Convlutional Neural Network (CNN): tensorflow
- Gradient Boost : catboost, xgboost, lightgbm
- Linear Regression: statsmodels
- RandomForestRegressor: scikit-learn

. . . More models will be added in the future releases . . .

2.2 Install

2.2.1 PyPI

```
pip install pymltoolkit
```

If the installation failed with dependancy issues, execute the above command with –no-dependencies

```
pip install pymltoolkit --no-dependencies
```

2.2.2 Setup TensorFlow with GPU support (Optional)

Refer the official TensorFlow documentation (https://www.tensorflow.org/install/gpu) for most up to date innstructions.

PyMLToolKit is tested with the following software versions in Windows 10

- CUDA Toolkit 10.0 (10.0.130_411.31_win10)
- cuDNN SDK (v7.4.2.24)

Step #1

- Install latest NVIDIA® GPU drivers
- Install CUDA Toolkit
- Install cuDNN SDK

- Set System Path to CUDA Toolkit. If the CUDA Toolkit is installed to "C:/Program Files/NVIDIA GPU Computing Toolkit/CUDA/v10.0" and extracted cuDNN content to r"C:/Program FilesNVIDIA GPU Computing Toolkit/cuDNN", update your %PATH% to match:

```
SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\bin;%PATH%
SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.
↪0\extras\CUPTI\libx64;%PATH%
SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.0\include;%PATH%
SET PATH=C:\Program Files\NVIDIA GPU Computing Toolkit\cuDNN\bin;%PATH%
```

Step #2

- Install TensorFLow-GPU

PyPI

```
pip install tensorflow-gpu
```

To install specific version

```
pip install tensorflow-gpu==1.14
```

- Check GPU in Tensorflow (output forat as below)

```
from tensorflow.python.client import device_lib
print(device_lib.list_local_devices())
```

```
[name: "/device:CPU:0"
device_type: "CPU"
memory_limit: 99999999
locality {
}
incarnation: 9999999999, name: "/device:GPU:0"
device_type: "GPU"
memory_limit: 99999999
locality {
   bus_id: 1
   links {
   }
}
incarnation: 99999999
physical_device_desc: "device: 0, name: XXXXXX, pci bus id: 0000:00:00.0, compute␣
↪capability: 0.0"]
```

memory_limit is in bytes. To convert allocated memeory to GB use : memory_limit/(1024*1024*1024)

If you encounter errors in setting up TensorFlow, please refer to thw official TensorFlow Build and install error messages (https://www.tensorflow.org/install/errors)

CHAPTER 3

Indices and Tables

- genindex
- modindex
- search