**PyML Documentation**

 Developed by Joohwan Lee

* PyML is developed in order to make Machine Learning accessible for everyone
* With the simplest step, but full specifications, PyML delivers maximum efficiency
* Built on NumPy, matplotlib, scikit-learn, PyTorch, Seaborn
* What can you do with PyML?
* Data conversion
* Data validation test
* Data preprocessing
* Machine learning algorithms
* Deep learning with hyperparameter tuning
* Optimize accuracy with data modification
* Data visualization
* PyML is beginner friendly, yet powerful enough for real world application

**Index**

1. Requirements
2. Intro to Machine Learning
3. DataProcessing
4. MachineLearning
5. Optimizer
6. OptimumAlgorithm

**Requirements**

pip install numpy

pip install matplotlib

pip install scikit-learn

pip install seaborn

pip install pandas

# CUDA 10.2

pip install torch==1.8.0+cu102 torchvision==0.9.0+cu102 torchaudio===0.8.0 -f https://download.pytorch.org/whl/torch\_stable.html

# CUDA 11.1

pip install torch==1.8.0+cu111 torchvision==0.9.0+cu111 torchaudio===0.8.0 -f https://download.pytorch.org/whl/torch\_stable.html

# CPU

pip install torch==1.8.0+cpu torchvision==0.9.0+cpu torchaudio===0.8.0 -f <https://download.pytorch.org/whl/torch_stable.html>

pip install pyml

**Intro to Machine Learning**

* PyML consists of machine learning algorithms based on scikit-learn and PyTorch
* Major steps of Machine Learning

1. Find dataset with enough information that has correlations with the target.
2. Select an algorithm that might be efficient for your goals
3. Preprocess dataset in a way that it can suit your desired algorithms
4. Apply your dataset in an algorithm and test out the accuracy
5. Tune your dataset or hyperparameters to get the optimum model
6. Use the model to predict labels

* PyML offers functions that can optimize the model and dataset for you

**DataProcessing**

* ***args : filepath***
* *type(filepath)* : str
* *filepath* : File path of a dataset

**Methods**

csv\_to\_DataFrame(self, empty)

* ***args : drop***
* *type(drop)* : bool
* *drop* : Bool that determines to drop NaN values or fill with 0
* Converts csv file to Pandas DataFrame

***return df***

df\_to\_dict(self, df)

* ***args : df***
* *type(df)* : Pandas DataFrame
* *df* : DataFrame that is given for process
* Converts DataFrame into dictionary

***return fin\_dict***

to\_dummies(self, df, lst\_dummies)

* ***args : df, lst\_dummies***
* *type(df)* : Pandas DataFrame
* *type(lst\_dummies)* : list
* *df* : DataFrame that is given for process
* *lst\_dummies* : list containing columns to convert to dummy variables
* Converts data in columns to dummy variables

***return df***

to\_floats(self, df, lst\_floats)

* ***args : df, lst\_floats***
* *type(df)* : Pandas DataFrame
* *type(lst\_floats)* : list
* *df* : DataFrame that is given for process
* *lst\_dummies* : list containing columns to convert to floats
* Converts data in columns to floats

***return df***

feature\_label\_split(self, df, feature\_lst, label\_lst)

* ***args : df, feature\_lst, label\_lst***
* *type(df)* : Pandas DataFrame
* *type(feature\_lst)* : list
* *type(label\_lst)* : list
* *df* : DataFrame that is given for process
* *feature\_lst* : list containing head of columns to use for features
* *label\_lst* : list containing head of columns to use for label
* Divides DataFrame into feature DataFrame and label DataFrame

***return feature, label***

train\_test\_split(self, feature, label, train\_size=0.8, random\_state=None)

* ***args : df, feature, label, train\_size, random\_state***
* *type(df)* : Pandas DataFrame
* *type(feature)* : Pandas DataFrame
* *type(label)* : Pandas DataFrame
* *type(train\_size)* : float
* *type(random\_state)* : int
* *df* : DataFrame that is given for process
* *feature* : DataFrame containing features
* *label* : DataFrame containing labels
* *train\_size* : Training size (default = 0.8)
* *random\_state* : Set random state (default = None)
* Divides feature and label data into test and train data

***return feature\_train, feature\_test, label\_train, label\_test***