Digit Recognizer Documentation

Release 0.1

Rafael Lopes Conde dos Reis

August 13, 2016

Contents

Contents:

CHAPTER 1

Kaggle - Digit Recognizer

Solutions for Kaggle's Digit Recognizer competition.

> "The goal in this competition is to take an image of a handwritten single digit, and determine what that digit is. As the competition progresses, we will release tutorials which explain different machine learning algorithms and help you to get started.
>
> The data for this competition were taken from the MNIST dataset. The MNIST ("Modified National Institute of Standards and Technology") dataset is a classic within the Machine Learning community that has been extensively studied. More detail about the dataset, including Machine Learning algorithms that have been tried on it and their levels of success, can be found at http://yann.lecun.com/exdb/mnist/index.html."

CHAPTER 2

Getting Started

Create an enviroment for the project using *virtualenv*. If you don't have it installed just run:

```
>>> $ pip install virtualenv
```

To create and activate an enviroment (called mnisnt) run:

```
>>> $ virtualenv mnisnt
>>> $ source mnisnt/bin/activate
```

Install general requirements by running:

```
>>> $ make requirements
```

Install TensorFlow following the turorial on their website

If you want to run the project notebooks istall Jupyter and Matplotlib by running:

```
>>> $ pip install jupyter
>>> $ pip install matplotlib
```

CHAPTER 3

Commands

The Makefile contains the central entry points for common tasks related to this project.

3.1 Data Preparation

make data will download train.csv and test.csv files to /data/raw/

3.2 Model Training

make train will train the model using train.csv and save to /models/

3.3 Making Predictions

make predict will run the model on test.csv data and save a CVF file with the predictions to /submissions/

CHAPTER 4

Project Structure

The project structure can be viewed bellow with some description:

```
-- LICENSE
-- Makefile           <- Makefile with commands like `make data` or `make train`
-- README.md          <- The top-level README for developers using this project.
-- SETTINGS.json      <- This file specifies the path to the train, test, model, and output
|                         directories.
-- data
|   -- interim        <- Intermediate data that has been transformed.
|   -- processed      <- The final, canonical data sets for modeling.
|   -- raw            <- The original, immutable data dump.
|
-- docs               <- A default Sphinx project; see sphinx-doc.org for details
|
-- models             <- Trained and serialized models, model predictions, or model summaries
|
-- notebooks          <- Jupyter notebooks. Naming convention is a number (for ordering),
|                         the creator's initials, and a short `-` delimited description, e.g.
|                         `1.0-jqp-initial-data-exploration`.
|
-- requirements.txt   <- The requirements file for reproducing the analysis environment, e.g.
|                         generated with `pip freeze > requirements.txt`
|
-- src                <- Source code for use in this project.
|   -- __init__.py    <- Makes src a Python module
|   |
|   -- data           <- Scripts to download or generate data
|   |   -- make_dataset.py
|   |
|   -- features       <- Scripts to turn raw data into features for modeling
|   |   -- build_features.py
|   |
|   -- models         <- Scripts to train models and then use trained models to make
|       |                 predictions
|       -- predict.py
|       -- train.py
|
-- submissions        <- Kaggle submissions
|
-- tox.ini            <- tox file with settings for running tox; see tox.testrun.org
```