Python Keras Application Requirements

# High-level design

1. Project setup
   1. The user should be prompted to open an existing project or start a new one.
      1. Projects are stored in the *projects* folder at the highest level within the application.
   2. The user should be prompted for a source for the data.
      1. This can be either from an online source or local by the user.
      2. In the case the user chooses local the system should check if the *data* folder is not empty before proceeding.
   3. The user should be prompted for the nature of the project. Is it a Convolution Neural Network or a Recurrent Neural Network?
   4. A rudimentary model architecture should be defined in model.py
   5. User choices should be saved in a project meta file.
      1. Should the user wish to continue at a later time then the training progress will be saved.
2. Load the data set
   1. The system should be able to auto-detect the nature of the data: CSV file(s) or other separation methods, images (JPEG and PNG).
   2. From a sample of the data the system should define the input layer.
   3. The user should be prompted what the output layer should be, and the application will define this in model.py
3. Analyze the dataset
4. Define the model
5. Train the model
6. Evaluate trained model
7. Deploy trained model
8. Test projects
   1. The application should have several well documented test projects users can examine to quickly get under way. These should include.
      1. Neural Network using the Boston Housing Data.
      2. MNIST-fashion: with data from Zalando a basic CNN is created. This is considered an ‘Hello, world!’ example.
      3. Stock market data: used to create a simple Recurrent Neural network.
      4. Rest API: a simple coding example using ImageNet whose results can be accessed by REST API commands provided through the Python Flask library.