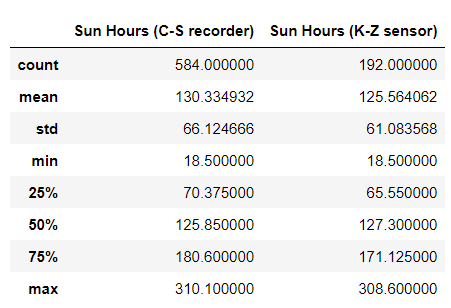
* Read data from MongoDB
  + London historical weather
  + New York City historical weather
* Variables
  + Inputs
    - London – tmin degF (monthly mean low temp), af days (monthly air frost days), rain inches ( total monthly rainfall), sun hours (total monthly sunshine duration)
    - New York City – tmin degF, monthly total precip, monthly total snowfall
  + Targets
    - London – tmax degF
    - New York City – tmax degF
* Clean data
  + Drop 1948 London data since af days is missing
  + Drop London data before 1957 since it’s missing sun hours
  + Decide what to do with sun hours, beginning 9/2005 the measuring devices for sun hours changed
    - Suggest we keep as is, (removing # from the later values measured by new sensors)
    - Overall measures seem consistent enough for our purposes
    - 
* Preprocessing
  + Are there categorical features? No
  + Split into inputs and output
    - X (inputs) – all other than tmax degF
    - y (output) – tmax degF
  + Split input data into training and testing datasets
  + Create StandardScaler instance
  + Train the scaler on training inputs
  + Scale the training and testing inputs
* Compile, train, and evaluate the model
  + Suggest we begin with the following:
    - Neural network
    - Two hidden layers, 6 neurons, 4 neurons, ReLU activation function
    - Output layer – linear activation function
  + Compile the model
    - model.compile(loss = ‘mse’, optimizer = ‘adam’, metrics=[tf.keras.metrics.MeanSquaredError()])
  + Training the model
    - Train the model on the X\_train\_scaled data
    - Suggest we start with 100 epochs
  + Evaluate the model
    - model\_loss, model\_mse = nn.evaluate(X\_test\_scaled, y\_test, verbose=2)
    - print(f’Loss: {model\_loss}, MSE: {model\_mse}’)
* Optimize our model performance as needed

For our ML model, a neural network was selected to take in the inputs (including minimum temperature, precipitation/snowfall, and, for London, air frost days and monthly sun hours). This model was selected in an attempt to leverage the flexibility of the various hidden layers, neurons, activation functions, etc.

The input data is scaled and our model is fitted on the training subset. Our model then predicts the monthly mean high temperature for the testing subset. The model is evaluated based upon loss and mean squared error (MSE).

We’re hopeful that our model can be used to present months that match a travelers desired high temperature.