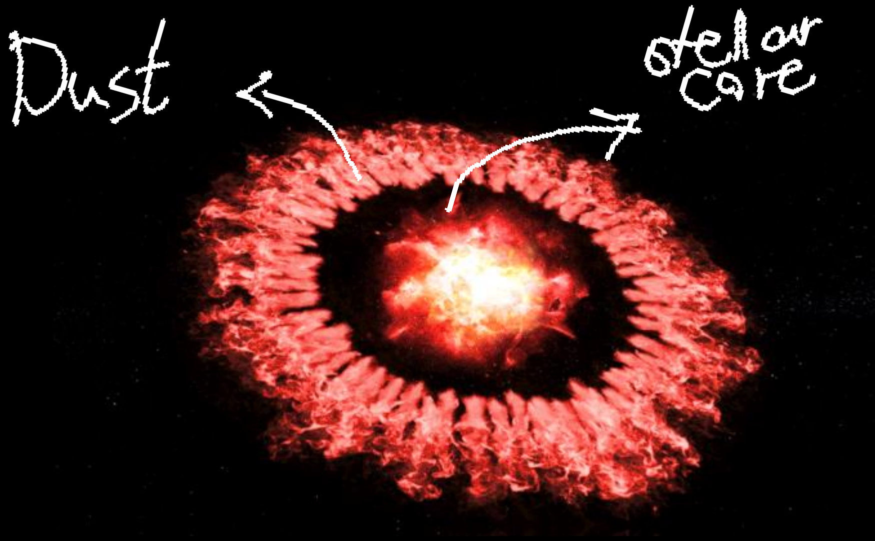
Supernova dust detection with neural networks

Zoe Ansari

Applied Machine Learning Course - NBI

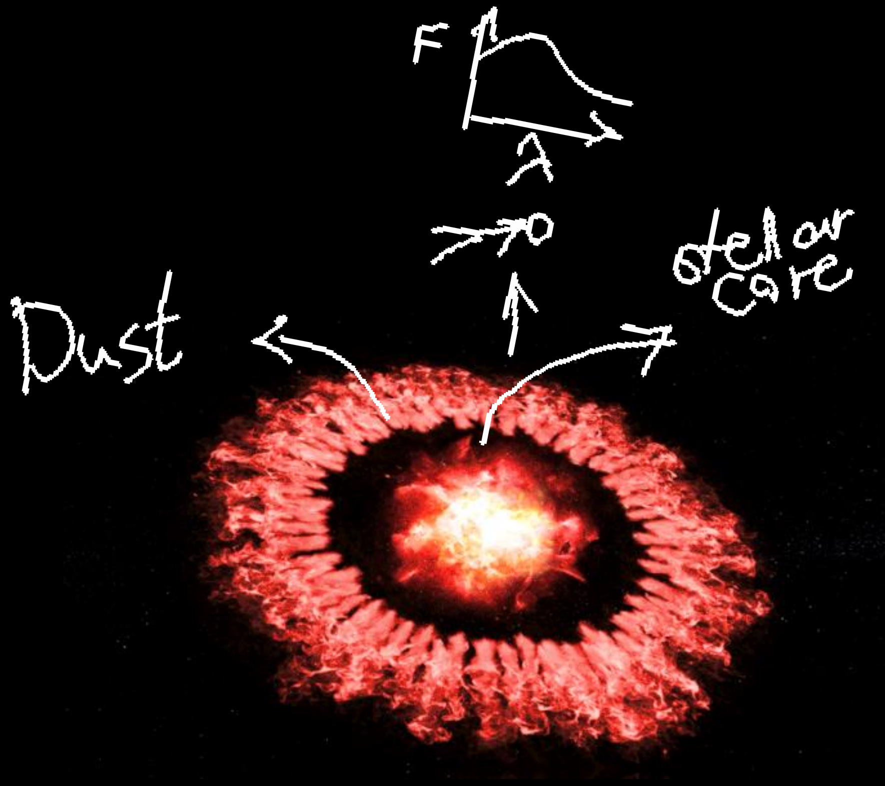
- Brief introduction on astrophysical aspects of the project
- Why dust?
- Data simulation
- Extracting desired features
- Try and tests on NN Model

Supernova

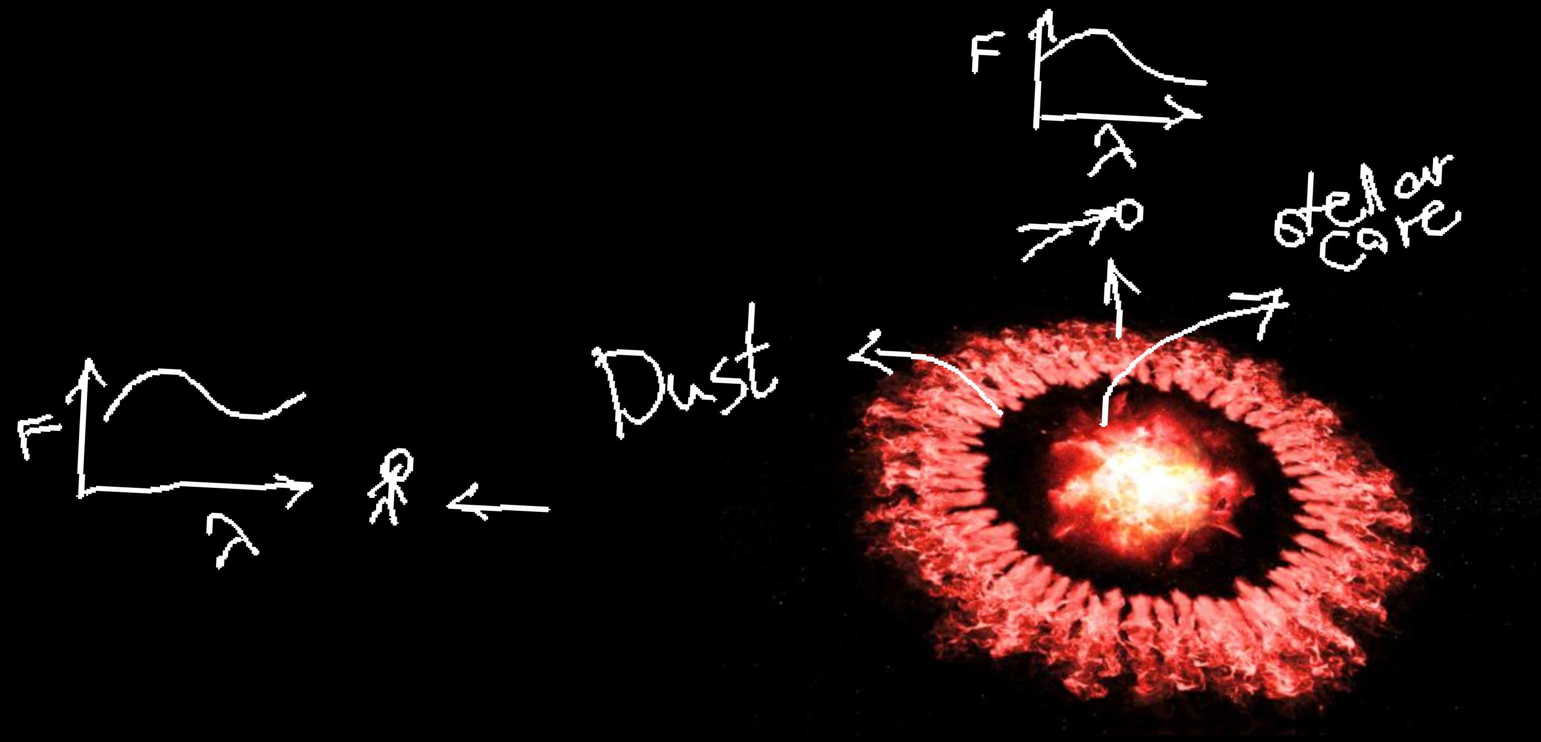


https://www.space.com/supernova-1987a-cosmic-dust-wake.html

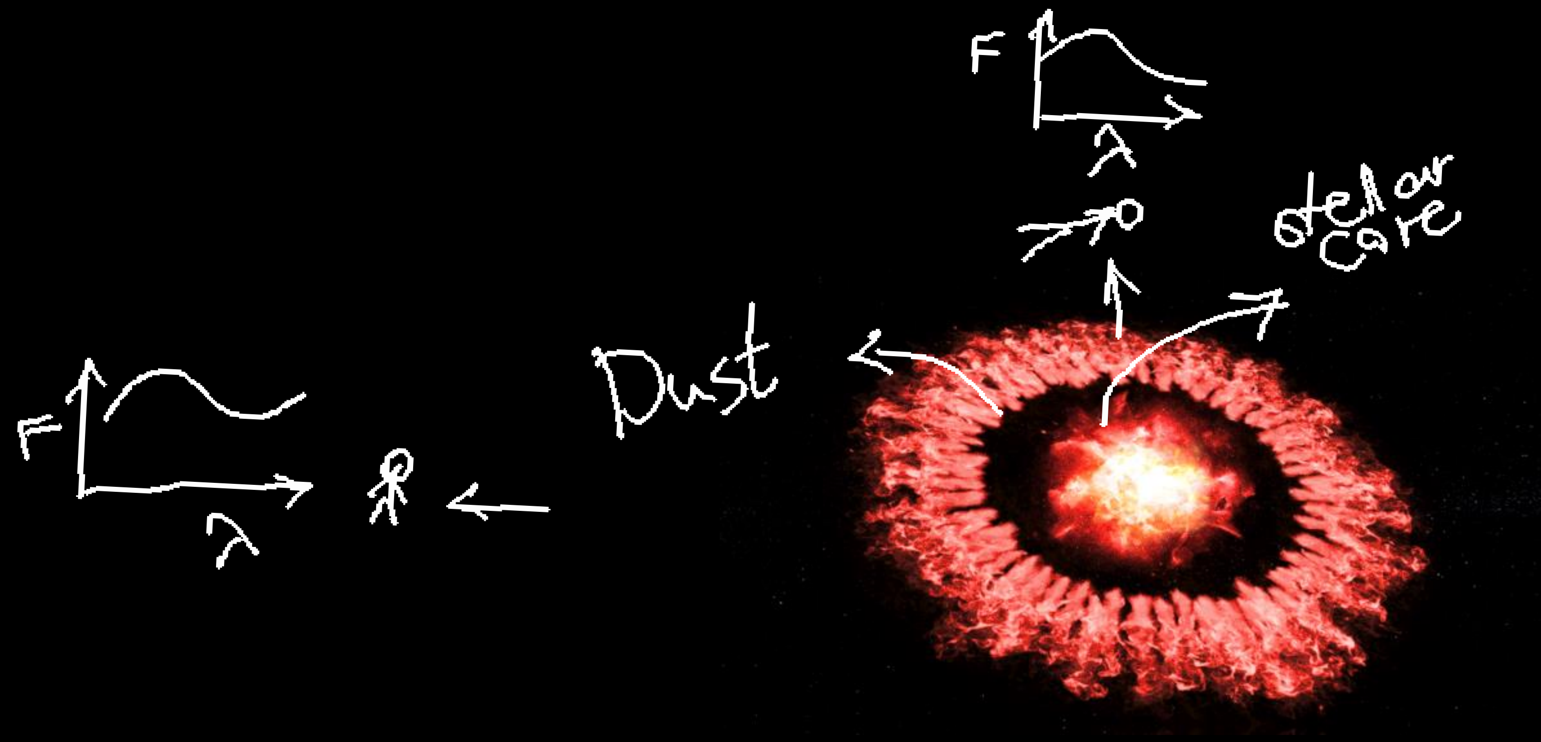
Supernova



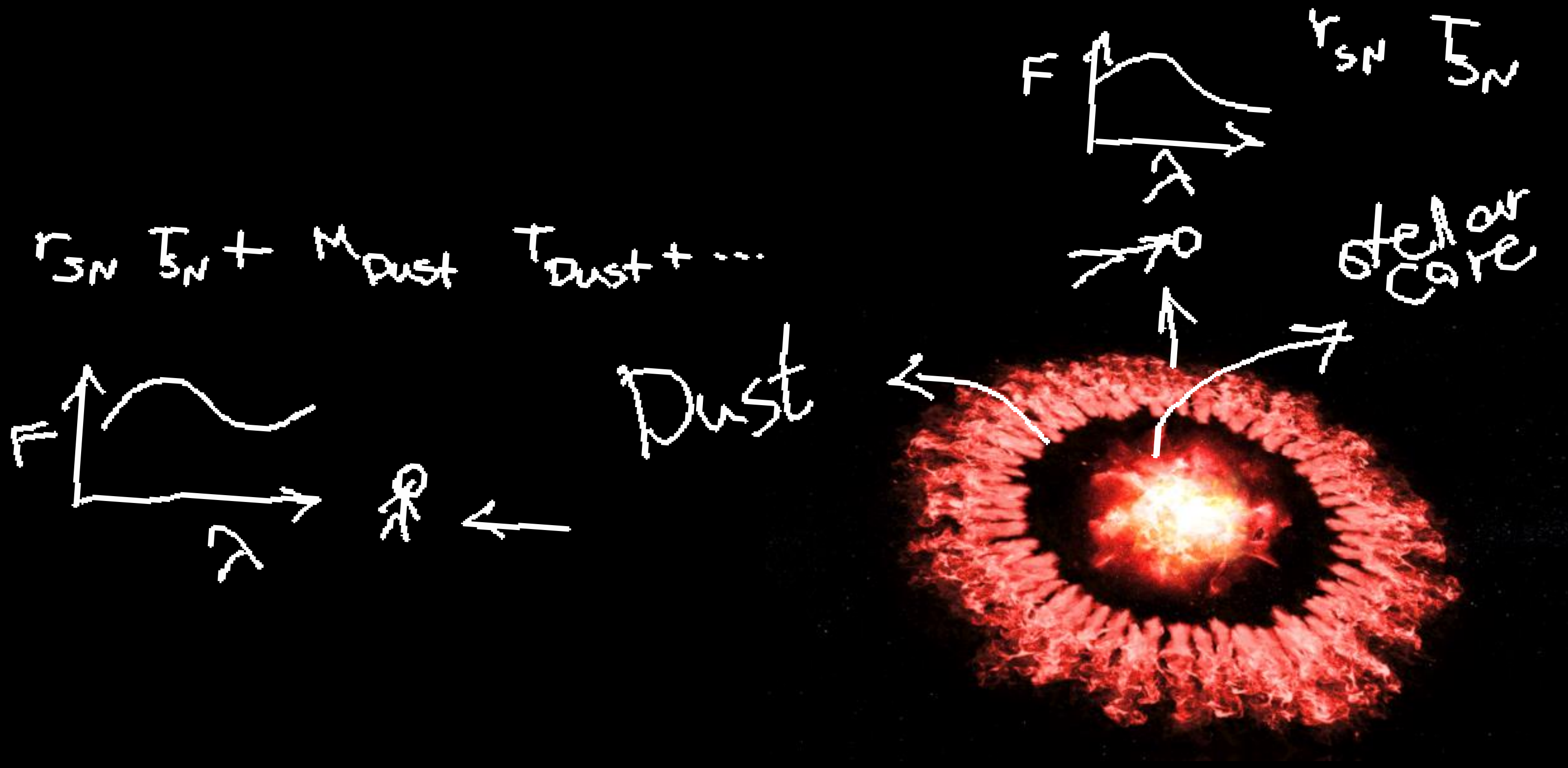
Why Dust?



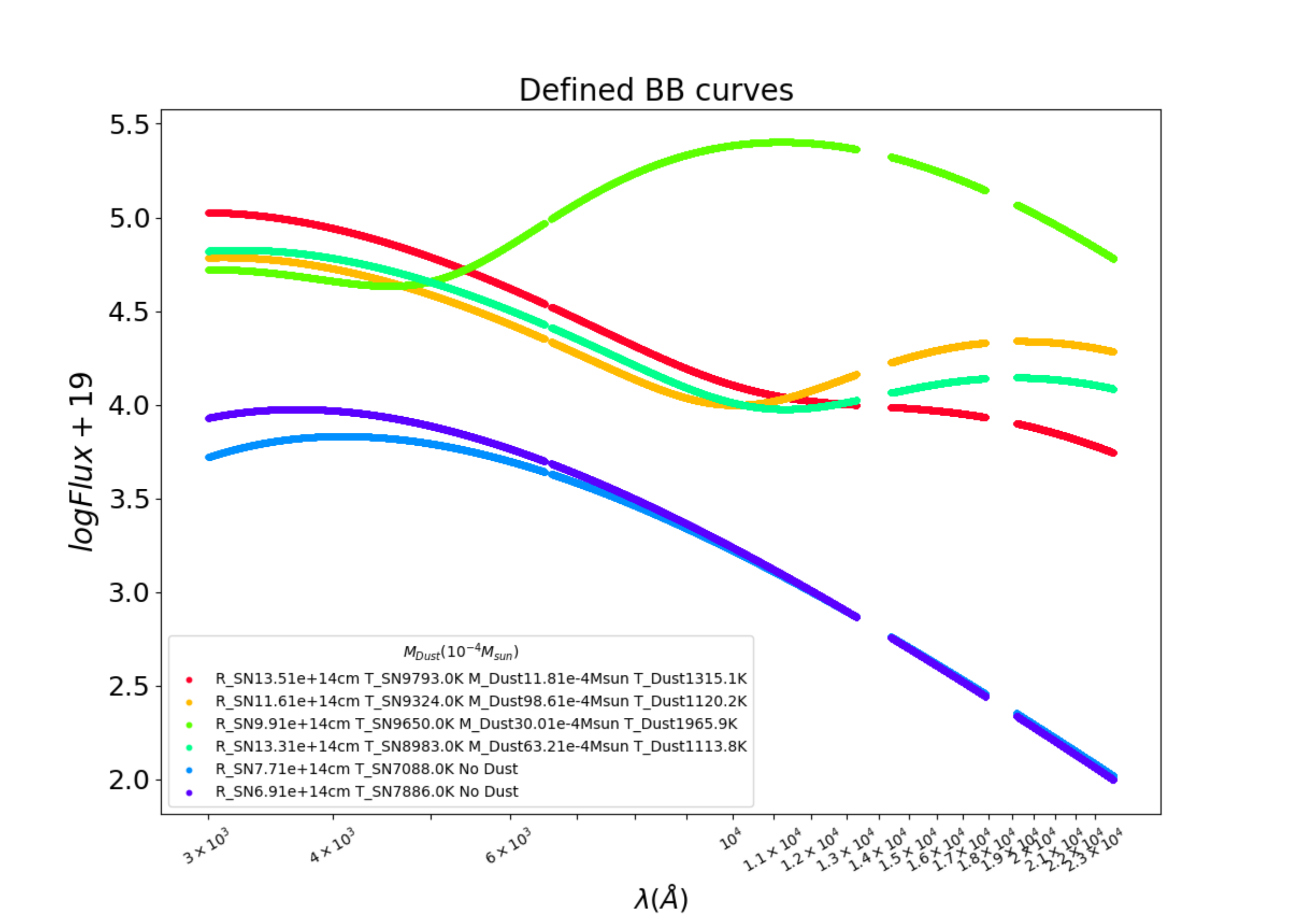
So long, and thanks for all the black bodies Planck



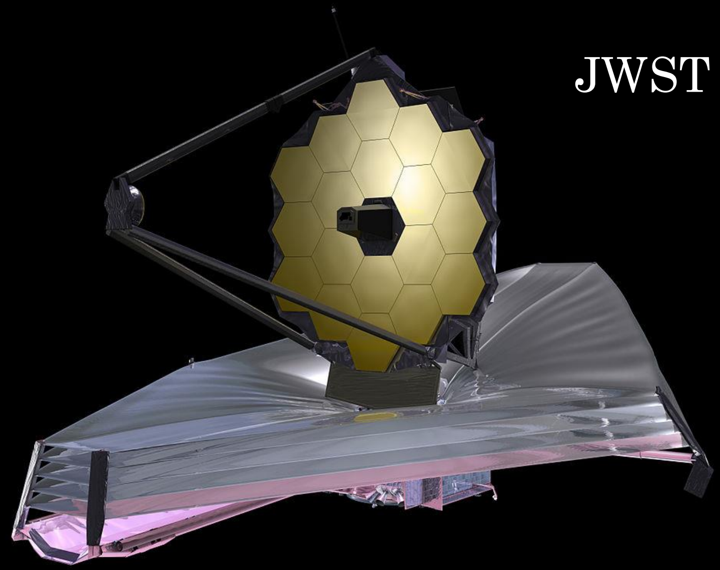
So long, and thanks for all the black bodies Planck



So long, and thanks for all the black bodies Planck

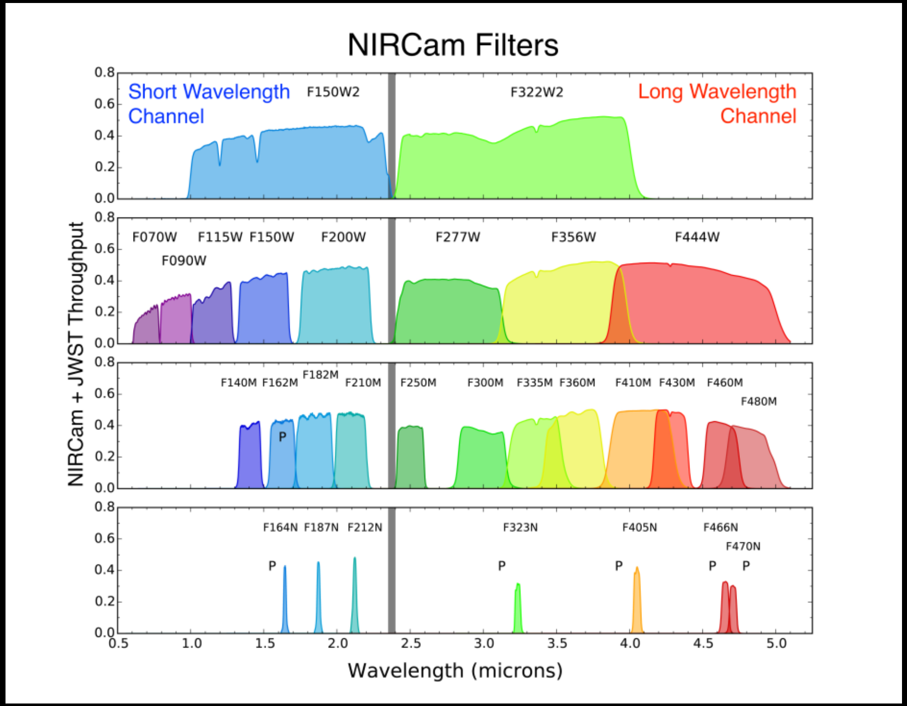


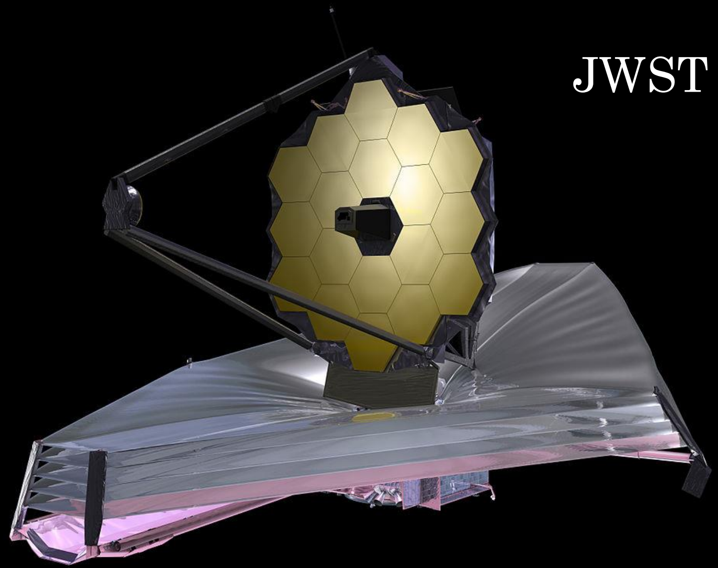
Photometric bands & the order of flux



JWST

Photometric bands & the order of flux

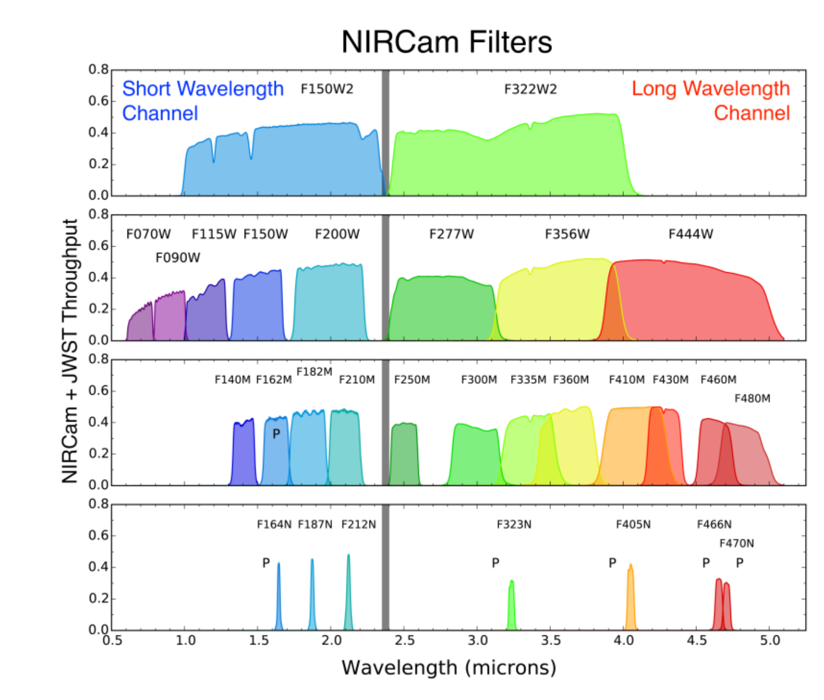


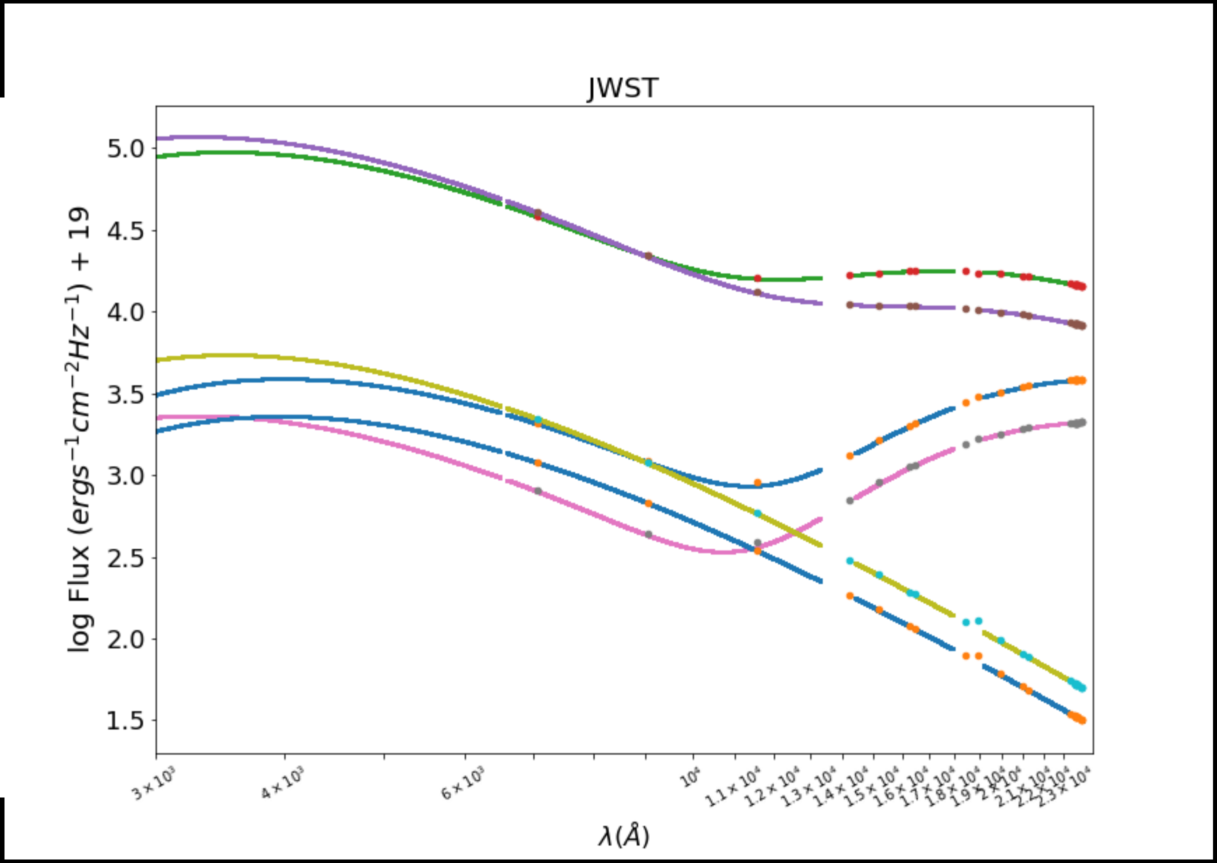


JWST

https://jwst-docs.stsci.edu/near-infrared-camera/nircam-instrumentation/nircam-filters

Photometric bands & the order of flux





https://jwst-docs.stsci.edu/near-infrared-camera/nircam-instrumentation/nircam-filters

Neural network begins

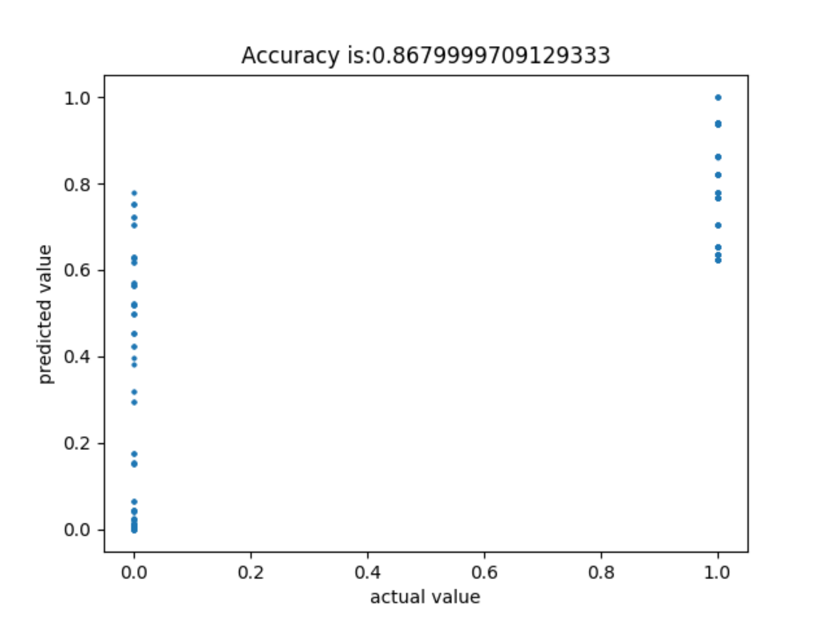
- Classification: identifying the existence of dust

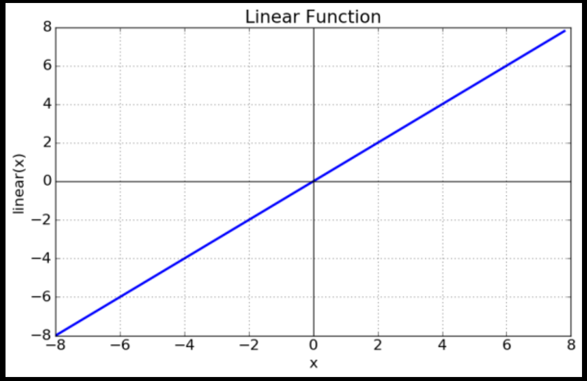
- Regression: estimating the amount of dust

- Classification: identifying the existence of dust

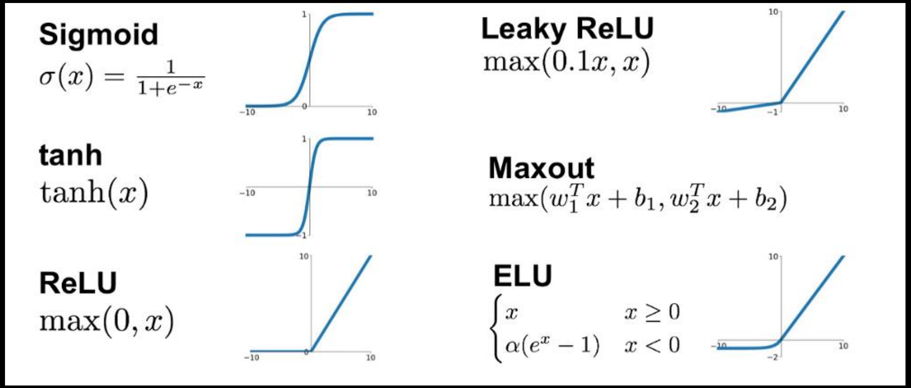
```
model.add(Conv1D(64, 1,input_shape=X_train.shape[1:]))
model.add(keras.layers.LeakyReLU(alpha=0.05))
model.add(Conv1D(filters=16, kernel_size=4))
model.add(keras.layers.LeakyReLU(alpha=0.05))
model.add(Conv1D(filters=32, kernel_size=1))#, activation='relu'))
model.add(keras.layers.LeakyReLU(alpha=0.05))
model.add(Dropout(rate=0.25))
model.add(Conv1D(filters=64, kernel_size=1))#, activation='relu'))
model.add(keras.layers.LeakyReLU(alpha=0.05))
model.add(Conv1D(filters=64, kernel_size=1))
model.add(keras.layers.LeakyReLU(alpha=0.05))
model.add(Dropout(rate=0.25))
model.add(Flatten())

model.add(Dense(1, activation='sigmoid'))
```



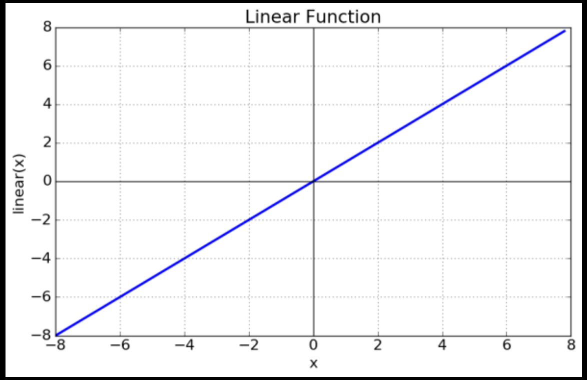


With the ame arch + last layer "linear"

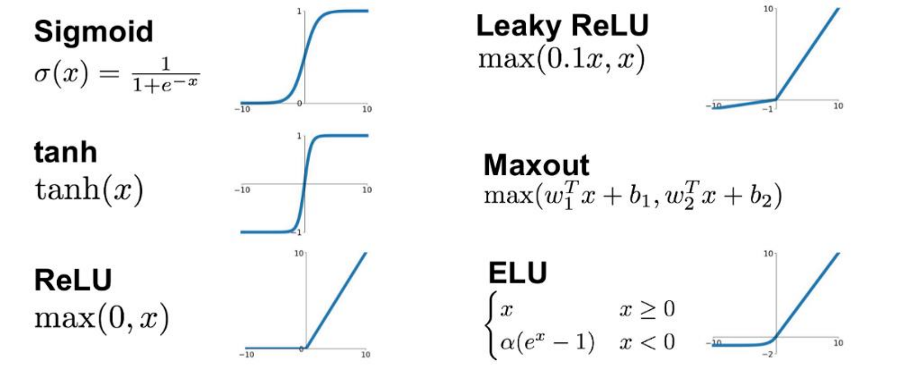

Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$
tanh
$\tanh(x)$
ReLU
$\max(0, x)$
Leaky ReLU
$\max(0.1x, x)$
Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$
ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

- Regression: estimating the amount of dust

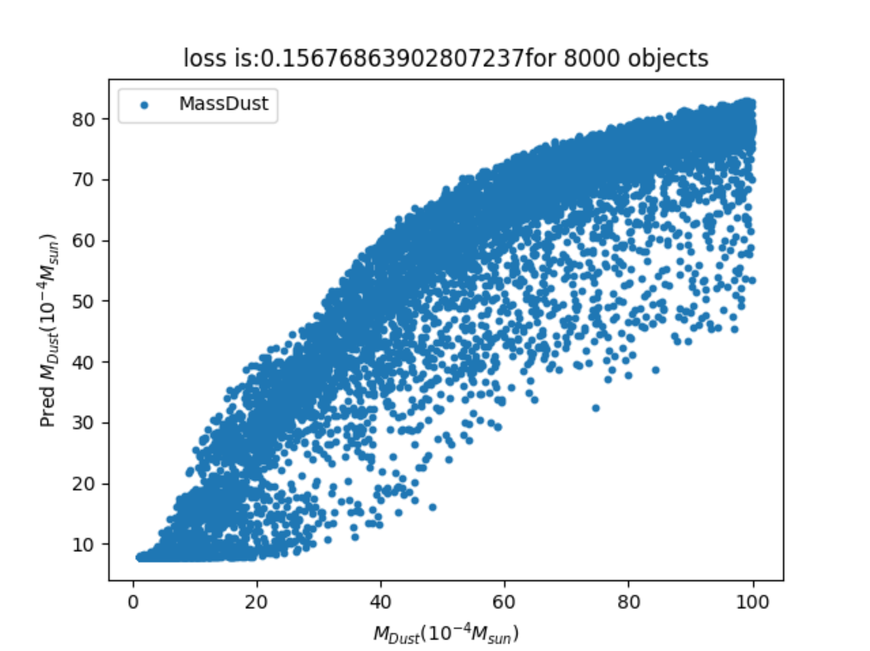
https://mc.ai/complete-guide-of-activation-functions/

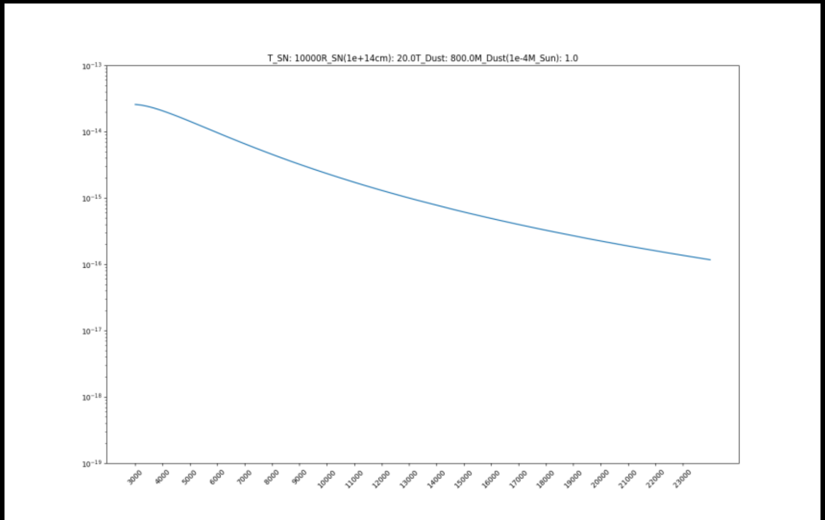


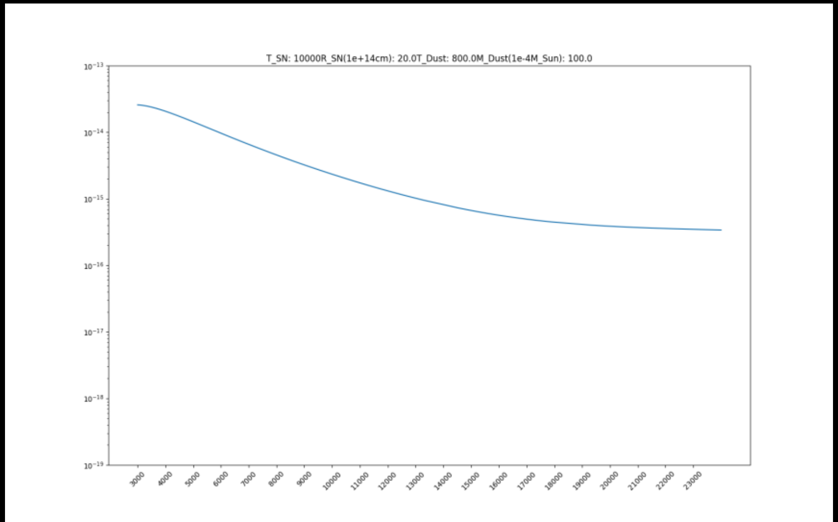
With the ame arch + last layer "linear"


Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$
tanh
$\tanh(x)$
ReLU
$\max(0, x)$
Leaky ReLU
$\max(0.1x, x)$
Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$
ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$

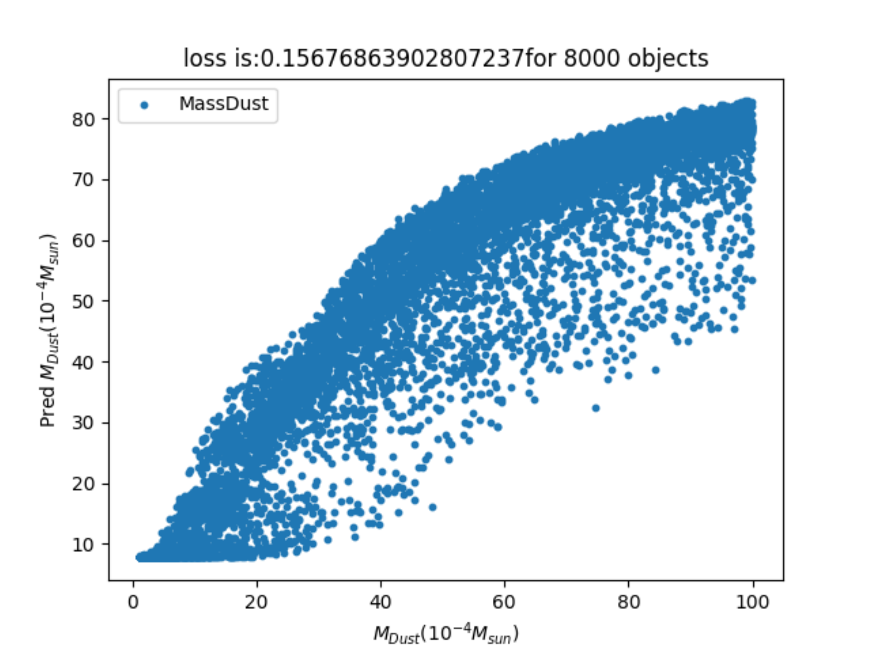
- Regression: estimating the amount of dust



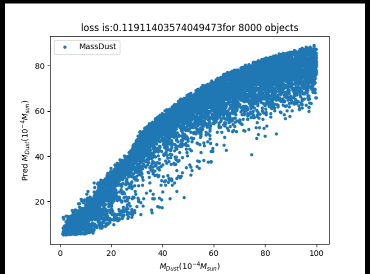




- Regression: estimating the amount of dust



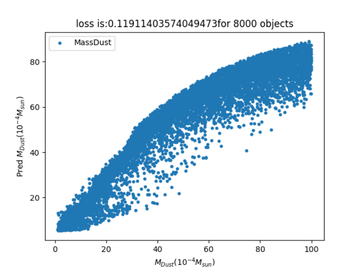
- Regression: estimating the amount of dust



$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_i^p)^2}{n}$$

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_i^p|}{n}$$

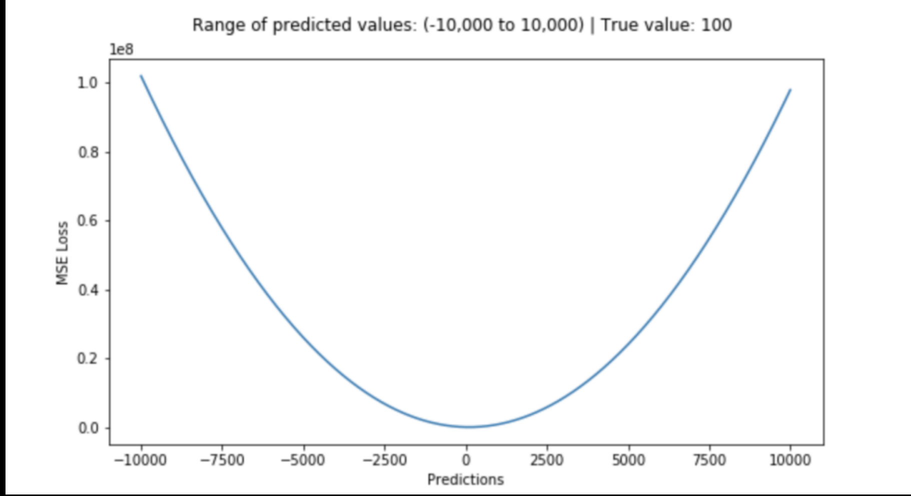
- Regression: estimating the amount of dust

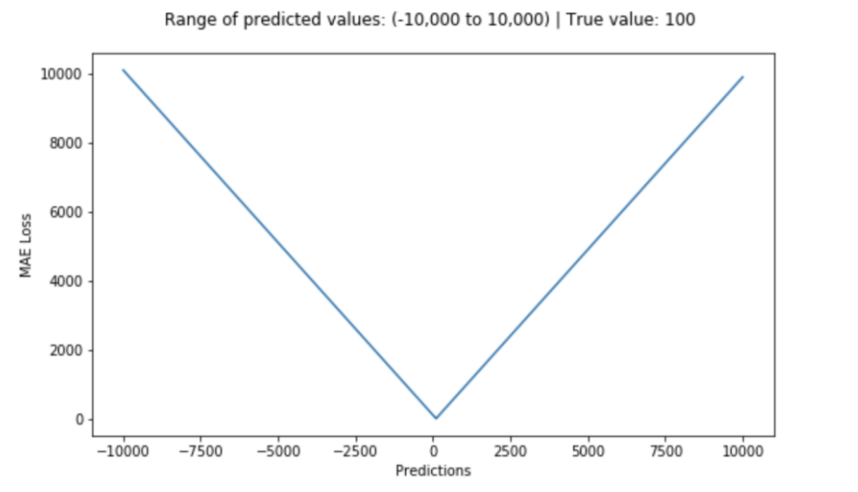


https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0

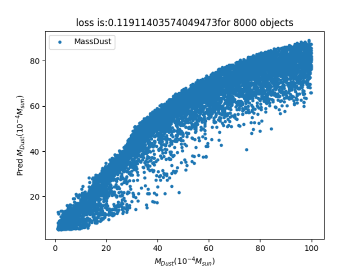
$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_i^p)^2}{n}$$

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_i^p|}{n}$$


Range of predicted values: (-10,000 to 10,000) | True value: 100

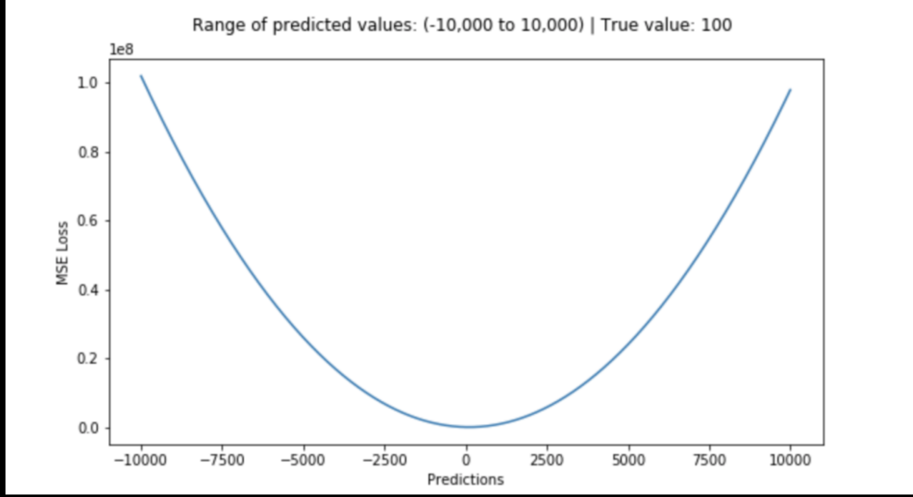

Range of predicted values: (-10,000 to 10,000) | True value: 100

- Regression: estimating the amount of dust

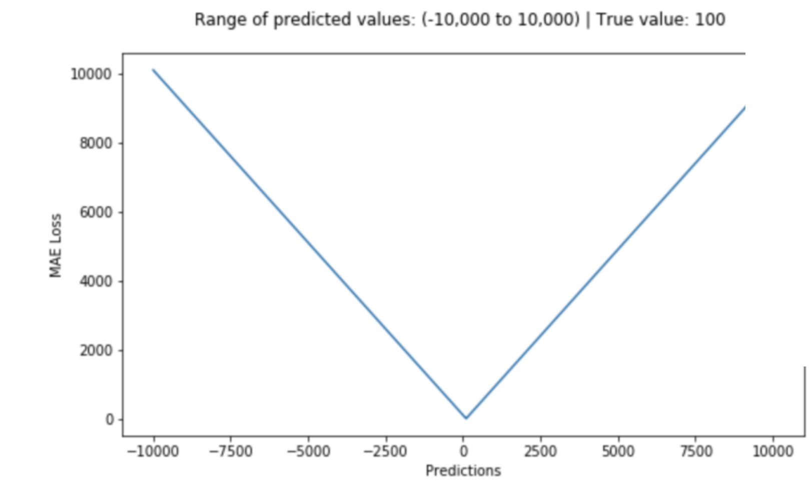

loss is:0.11911403574049473for 8000 objects

https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0

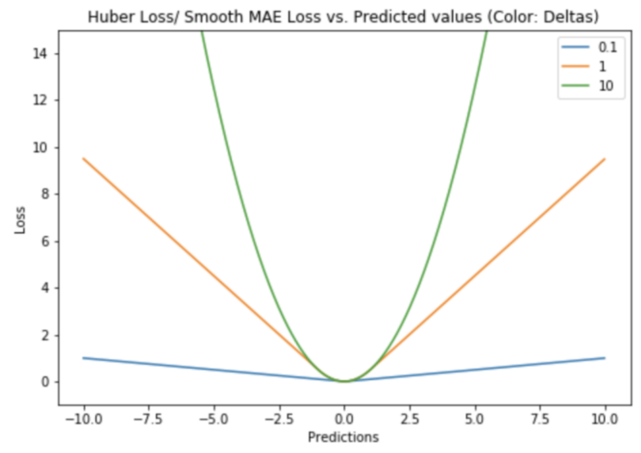
$$MSE = \frac{\sum_{i=1}^{n} (y_i - y_i^p)^2}{n}$$

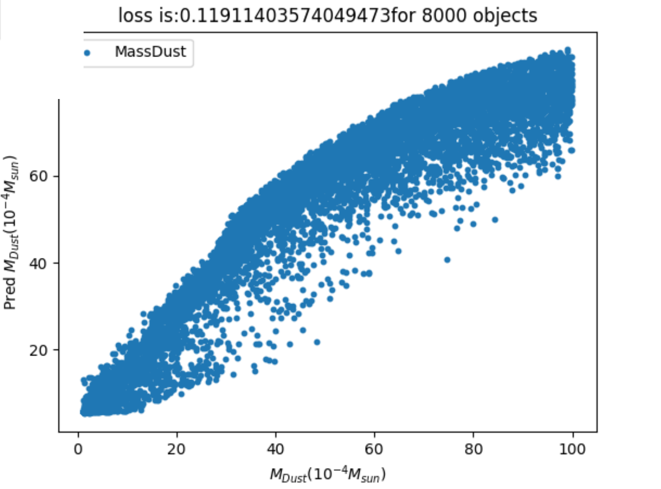

Range of predicted values: (-10,000 to 10,000) | True value: 100

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_i^p|}{n}$$


Range of predicted values: (-10,000 to 10,000) | True value: 100

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\, |y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$


Huber Loss/ Smooth MAE Loss vs. Predicted values (Color: Deltas)


loss is:0.11911403574049473for 8000 objects

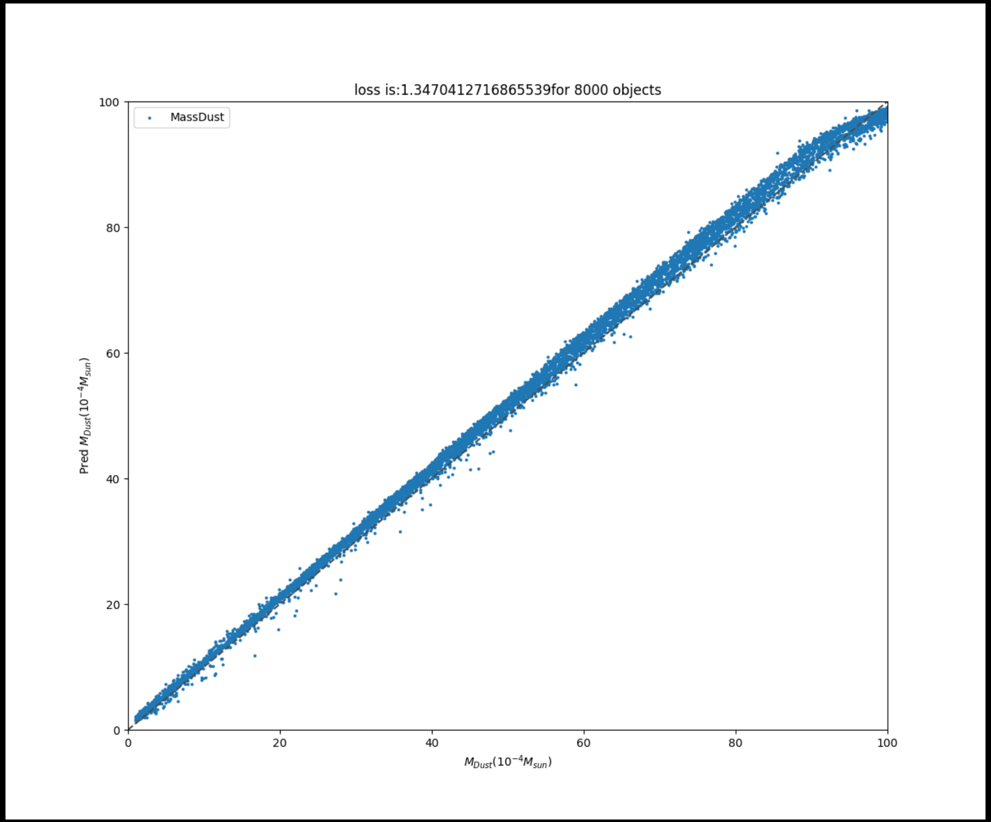
- Regression: estimating the amount of dust

https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0

Huber for 3000 epochs

+drop the dropouts

+Used PReLU instead of ReLUs

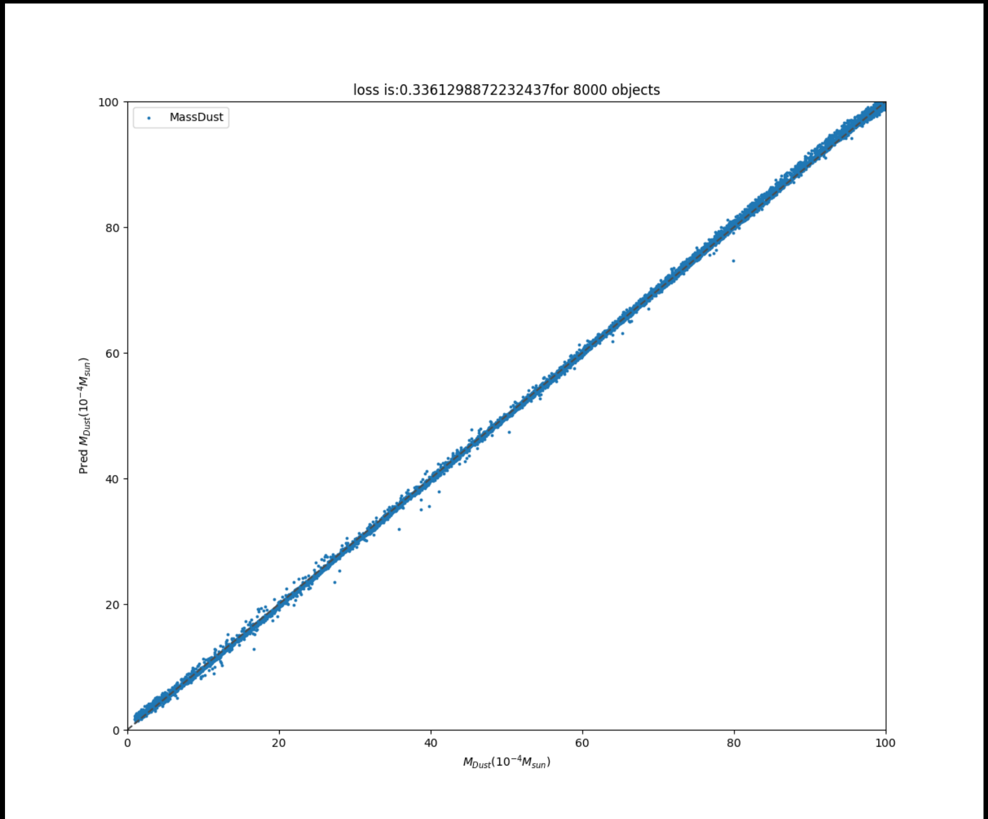


- Regression: estimating the amount of dust

Huber for **5000** epochs

+drop the dropouts

+Used PReLU instead of ReLUs



- Regression: estimating the amount of dust