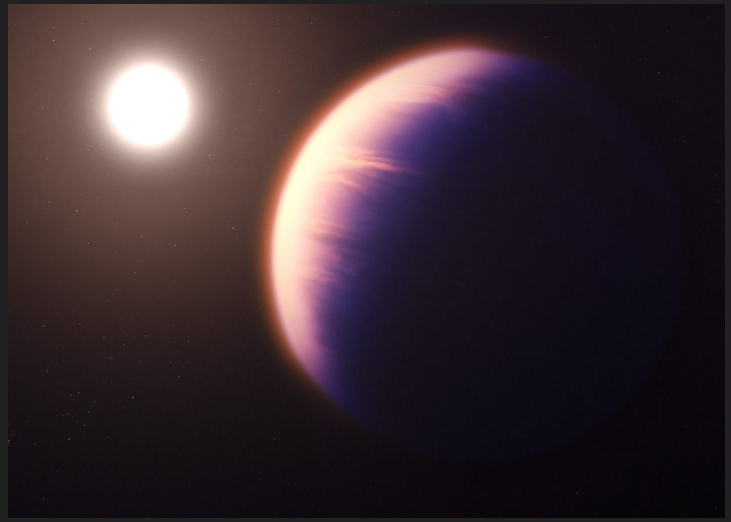
Exo-Planet Hunting

Using Deep-Learning ML Techniques

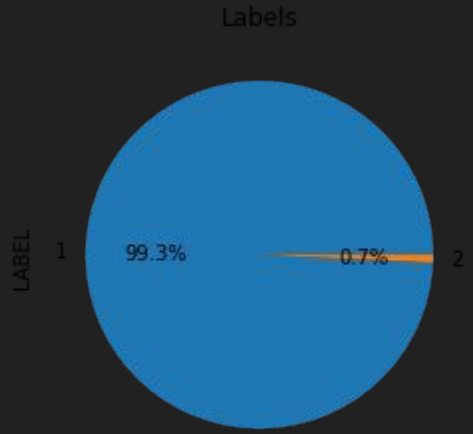
Dataset

- 6000+ stars
- 3197 Flux points
- Miluski Archive
  - MAST Home (stsci.edu)
  - K2 Mission - adding on to Kepler mission
- From NASA
- Find exoplanets using transit method



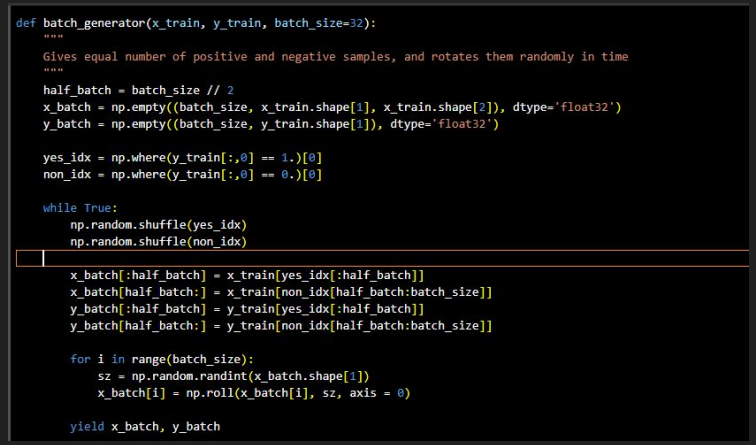
EDA and Visualization

- No missing values
- 3198 variables
  - 1 label and 3197 flux points
- Heavily Skewed
  - In line with domain knowledge
  - Cumming, Butler, et. al., "The Keck Planet Search: Detectability and the Minimum Mass and Orbital Period Distribution of Extrasolar Planets"
- Change to array



Splitting Data

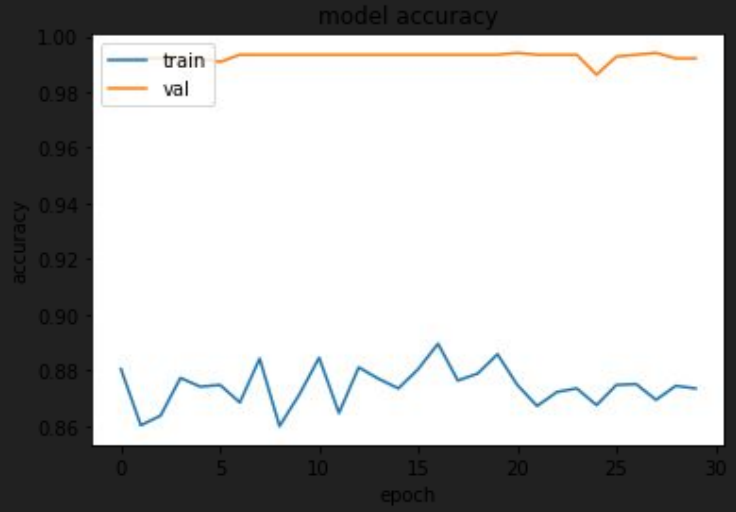
- Split use train_test_split
  - 7:3
  - Normalize the data
- Check train data set
- Use augmentation to “balance” the data
  - Create batch generator function



```
def batch_generator(x_train, y_train, batch_size=32):
    """
    Gives equal number of positive and negative samples, and rotates them randomly in time
    """
    half_batch = batch_size // 2
    x_batch = np.empty((batch_size, x_train.shape[1], x_train.shape[2]), dtype='float32')
    y_batch = np.empty((batch_size, y_train.shape[1]), dtype='float32')

    yes_idx = np.where(y_train[:,0] == 1.)[0]
    non_idx = np.where(y_train[:,0] == 0.)[0]

    while True:
        np.random.shuffle(yes_idx)
        np.random.shuffle(non_idx)

        x_batch[:half_batch] = x_train[yes_idx[:half_batch]]
        x_batch[half_batch:] = x_train[non_idx[half_batch:batch_size]]
        y_batch[:half_batch] = y_train[yes_idx[:half_batch]]
        y_batch[half_batch:] = y_train[non_idx[half_batch:batch_size]]

        for i in range(batch_size):
            sz = np.random.randint(x_batch.shape[1])
            x_batch[i] = np.roll(x_batch[i], sz, axis = 0)

        yield x_batch, y_batch
```

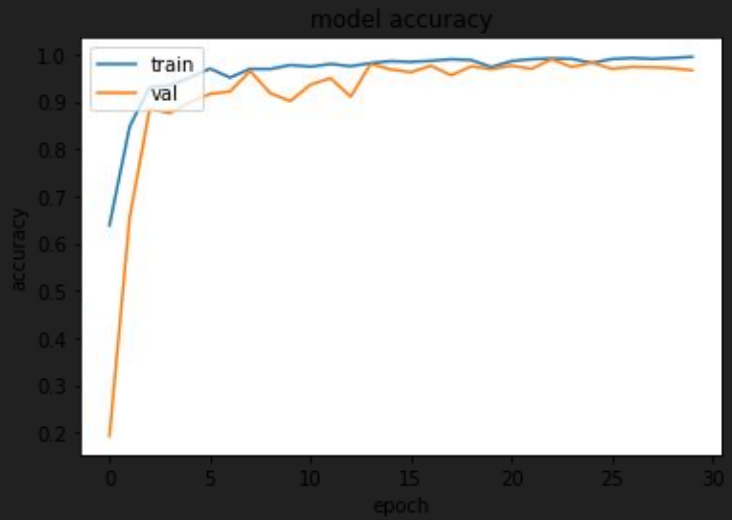
Model - 1

- Large number of Variables
- Based on research CNN is good
  - User previous Cancer
- Look at train and val acc
- 1D Conv
  - 3 X 2
  - Adam opt(.001)
- Too consistent
- High loss



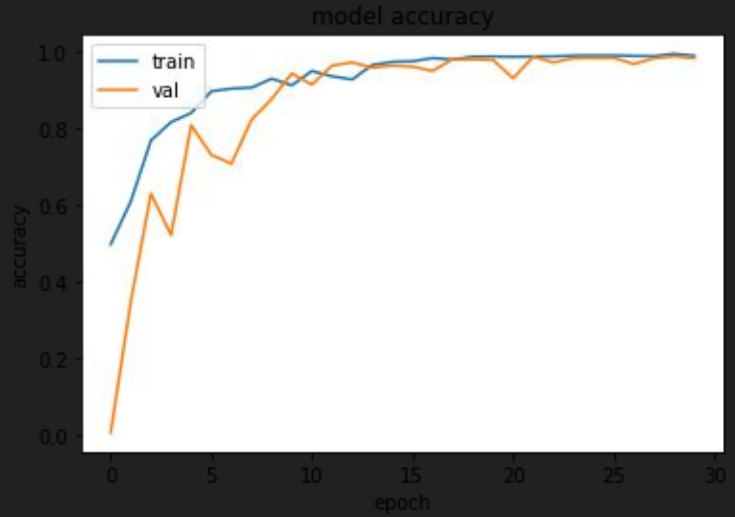
Model - 2

- 1D Conv
  - 4 X 1
  - Adam opt(.001)
- Much better loss
- See growth
- End with good val and train accuracy



Model - 3

- 1D Conv
  - 4 X 1
  - SGD
    - Default values
- Even better loss
- See growth
- End with good val and train accuracy
- Why SGD?
  - Trade-off of speed doesn't matter in this model



Conclusion

- Use third model to predict
- Cutoff around .97-.98
- 5 out of 6 points are labeled exoplanets
- Useful tool to go through thousands of data point
  - Human review like many other ML based tools
- Future iterations look at SGD
- ResNet or AlexNet