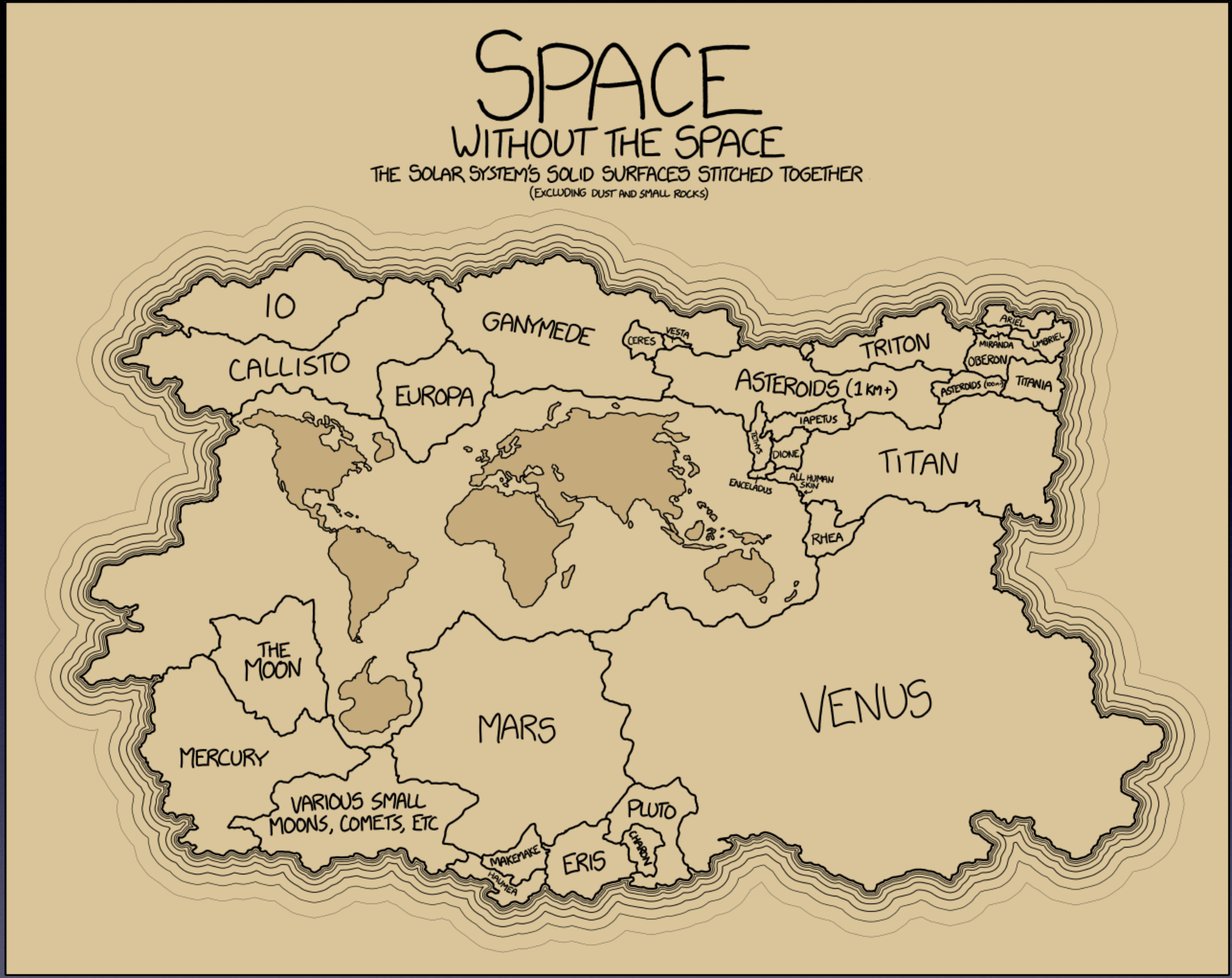
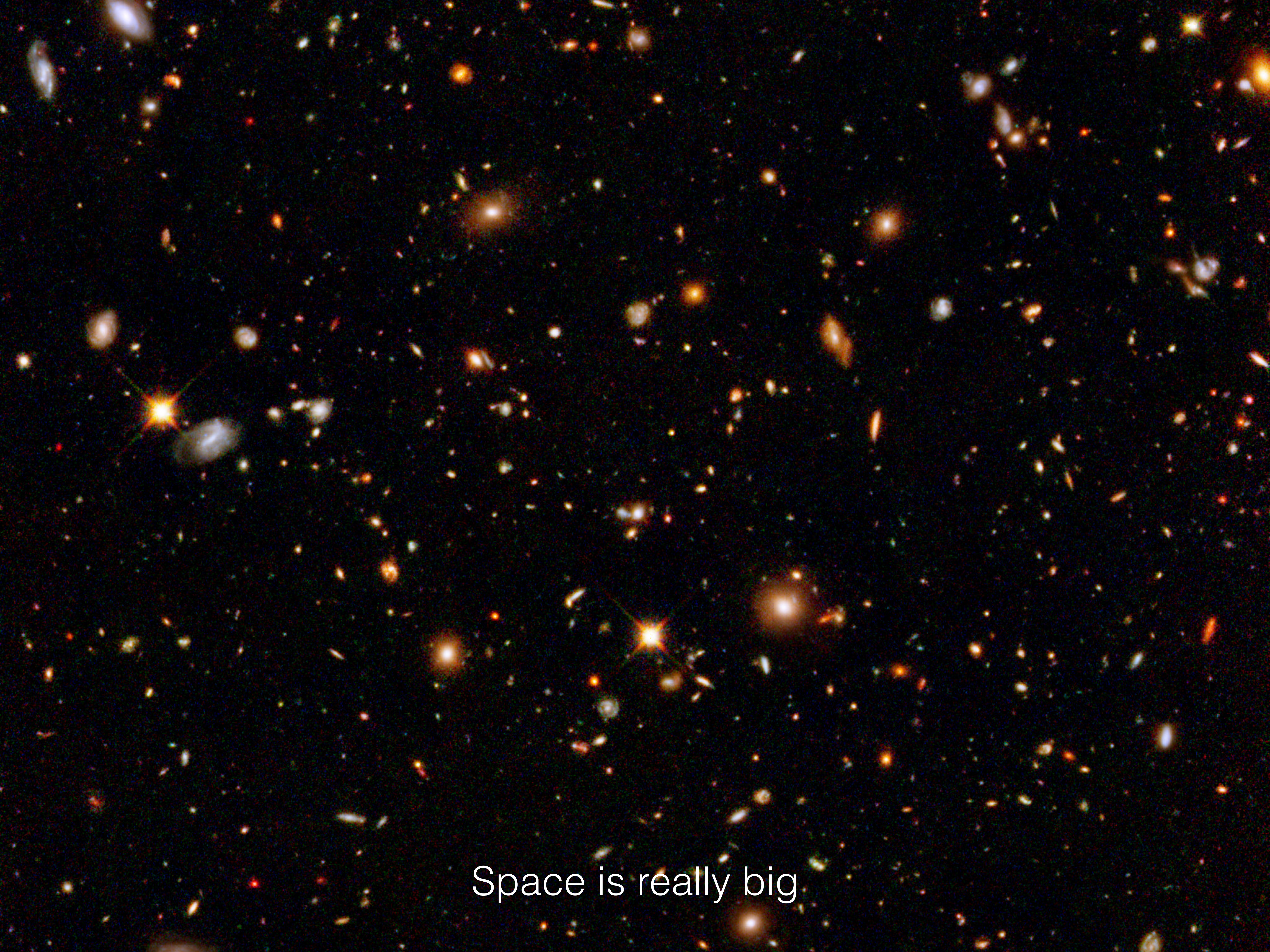
Detecting volcanos on Venus from SAR imagery taken by the Magellan spacecraft

Final Capstone - Andrew Boho



Why?

Space is big


Space is really big

Data growth in Astronomy

Growth driven by larger telescopes searching more of the sky (not just area, but different parts of the light spectrum)

- Galaxy Evolution Explorer space telescope produced (only) 20 terabytes from 2003-2012. Current space telescopes (James Webb and Kepler space telescopes, for example) are producing terabytes of data every night and petabytes per research project
- Large Synoptic Survey Telescope (Chile - 2019) will be equipped with a 3 billion pixel digital camera
- Adding the time dimension: exploring smaller portions of the sky but more frequently (Kepler exoplanet hunt)

Data

9,734 images

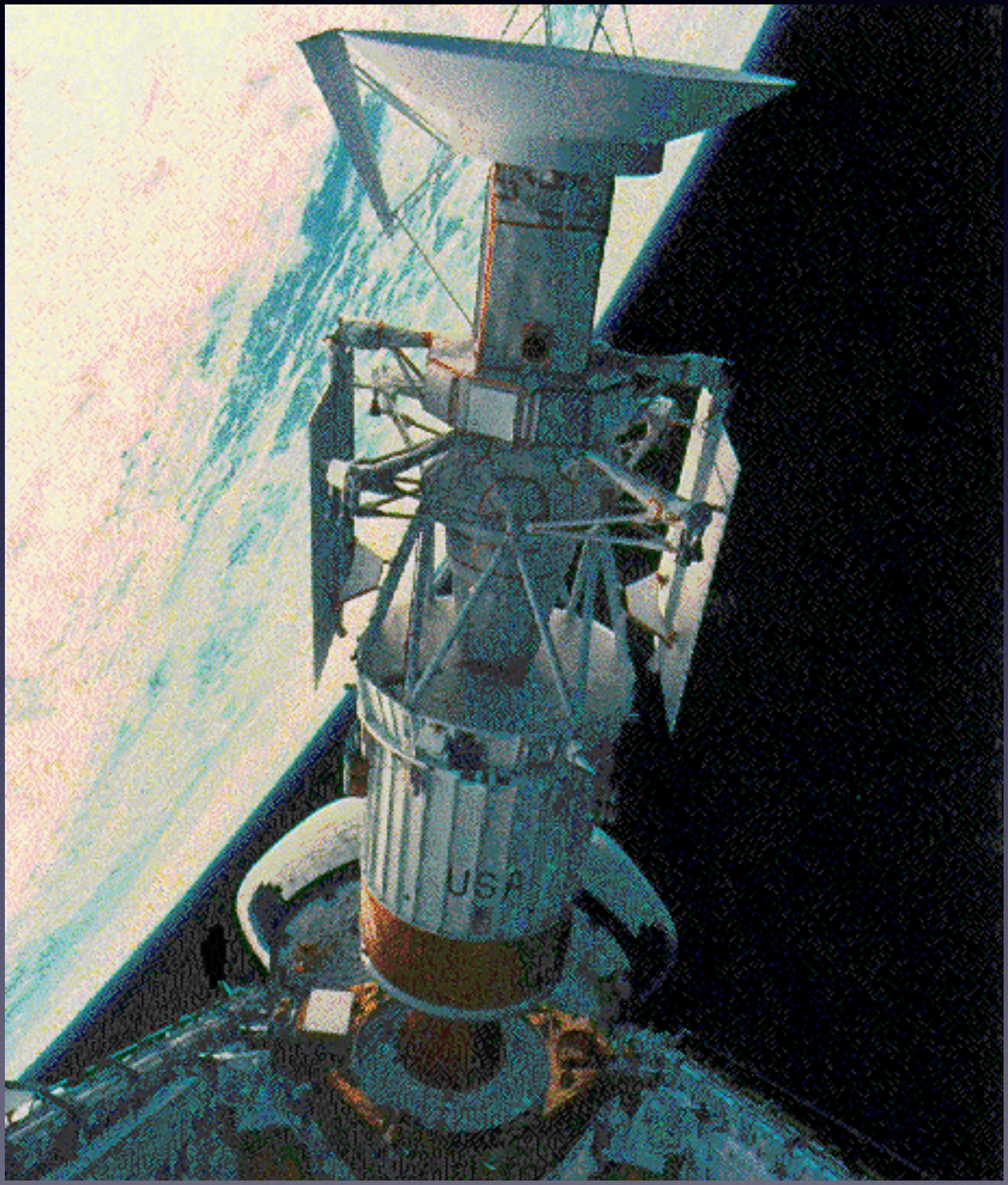
SAR - synthetic aperture radar images (think sonar)

Low resolution (110 x 110 pixels)

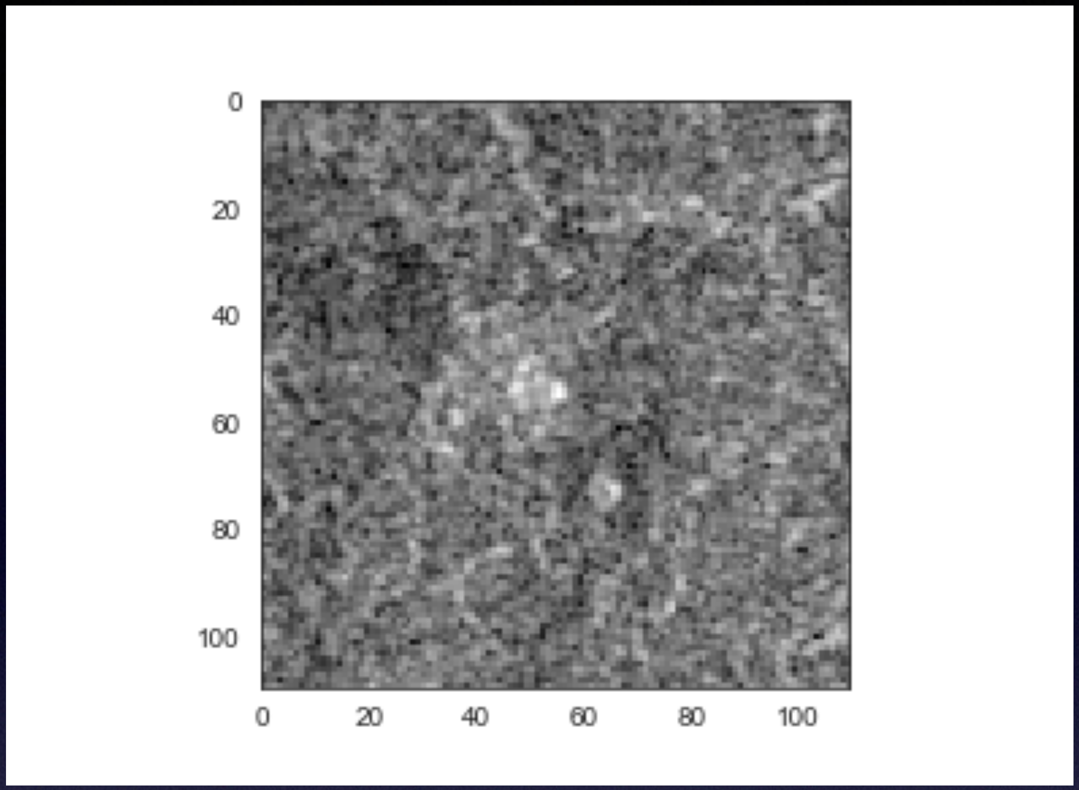
Grayscale with pixel values from 0 to 255

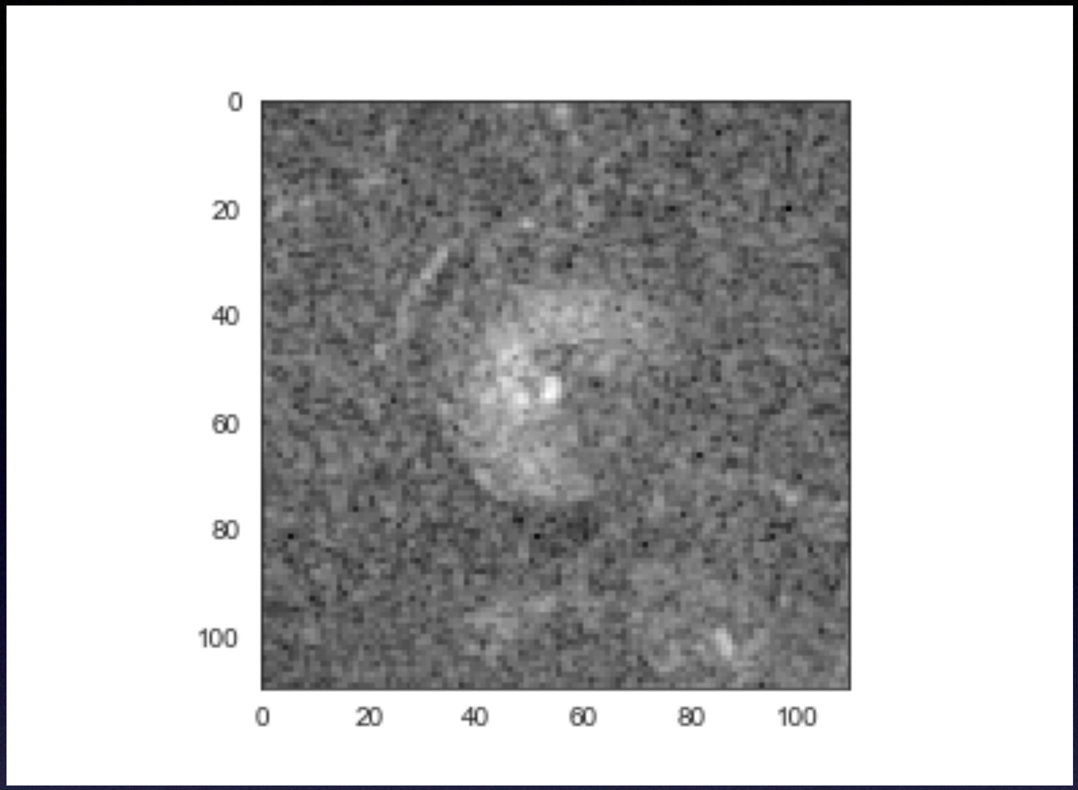
A matrix of 9,734 rows, and 12,100 columns

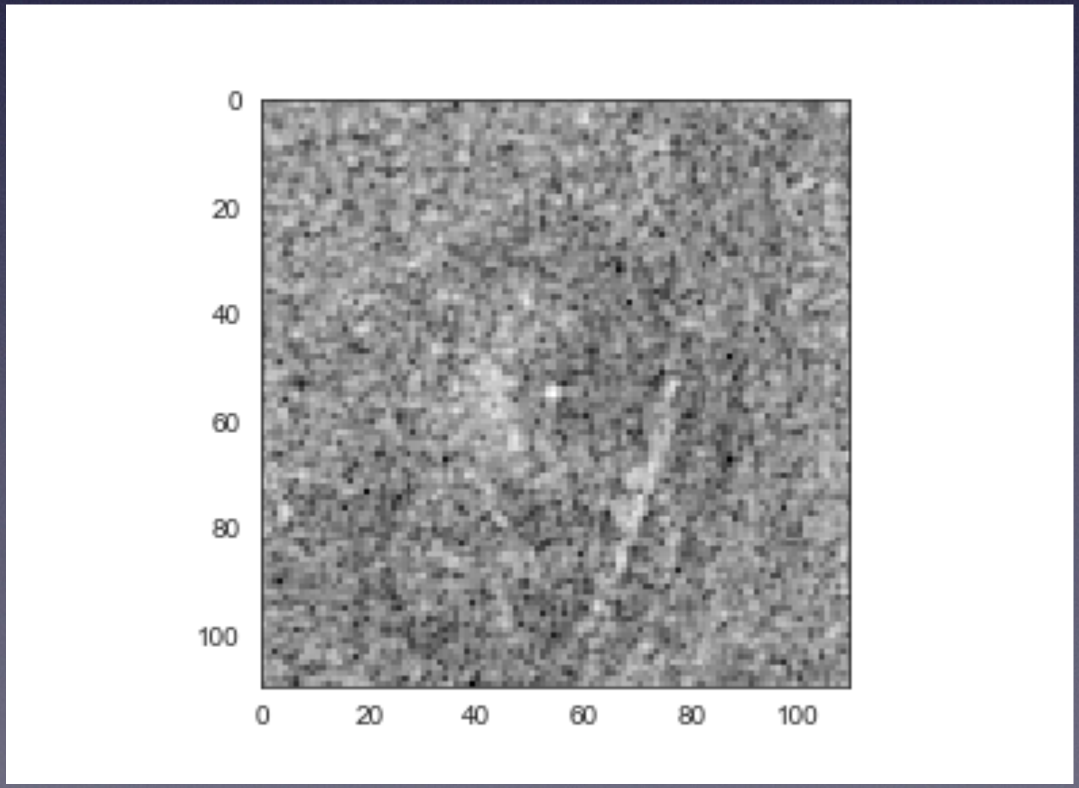
Each row is labeled as having a volcano in the image (1) or not having a volcano in the image (0)

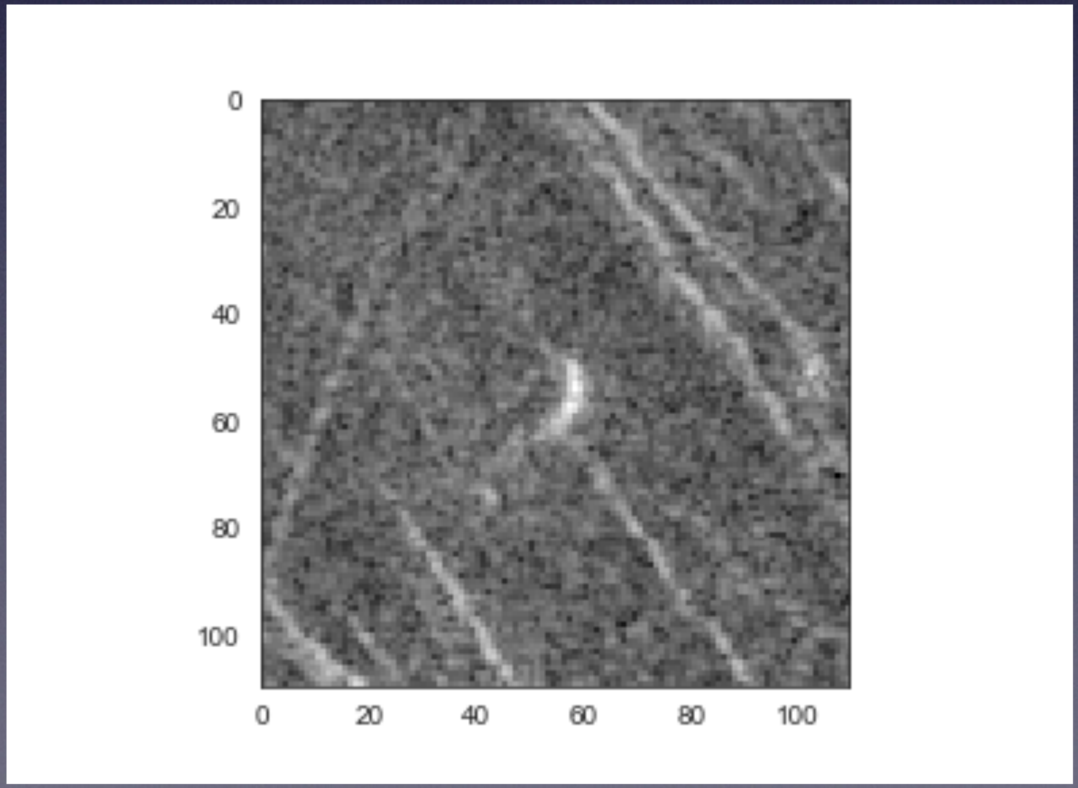


Space volcanos!

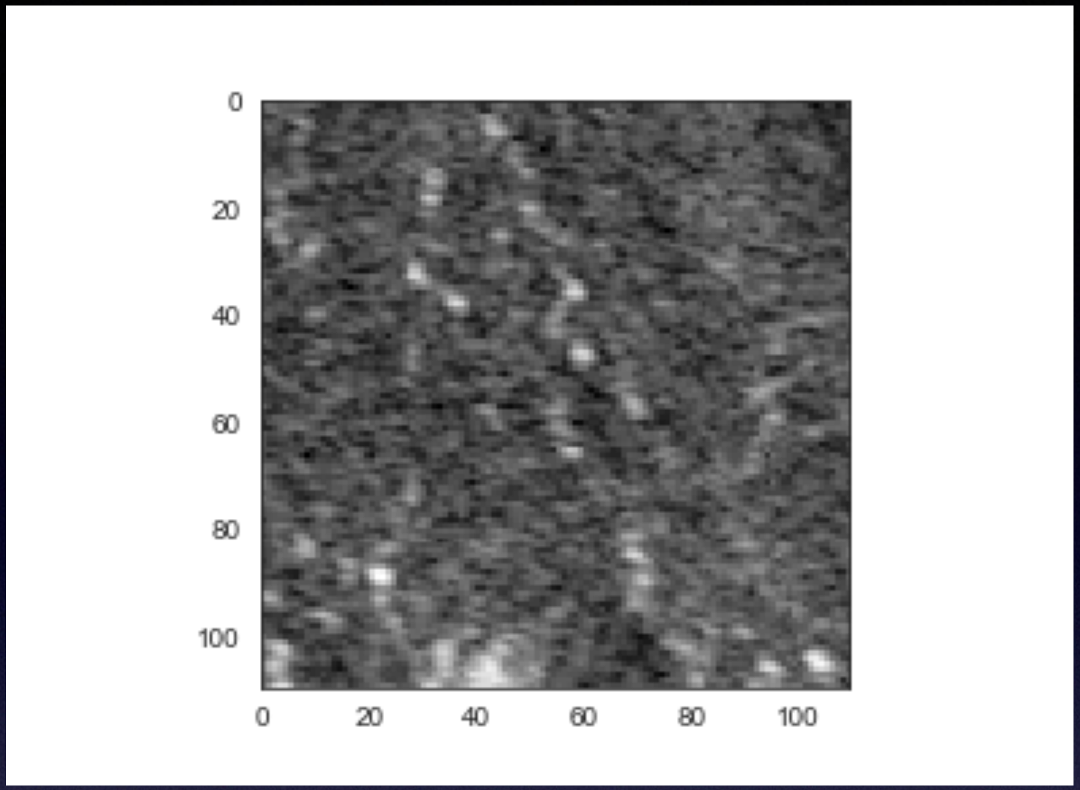


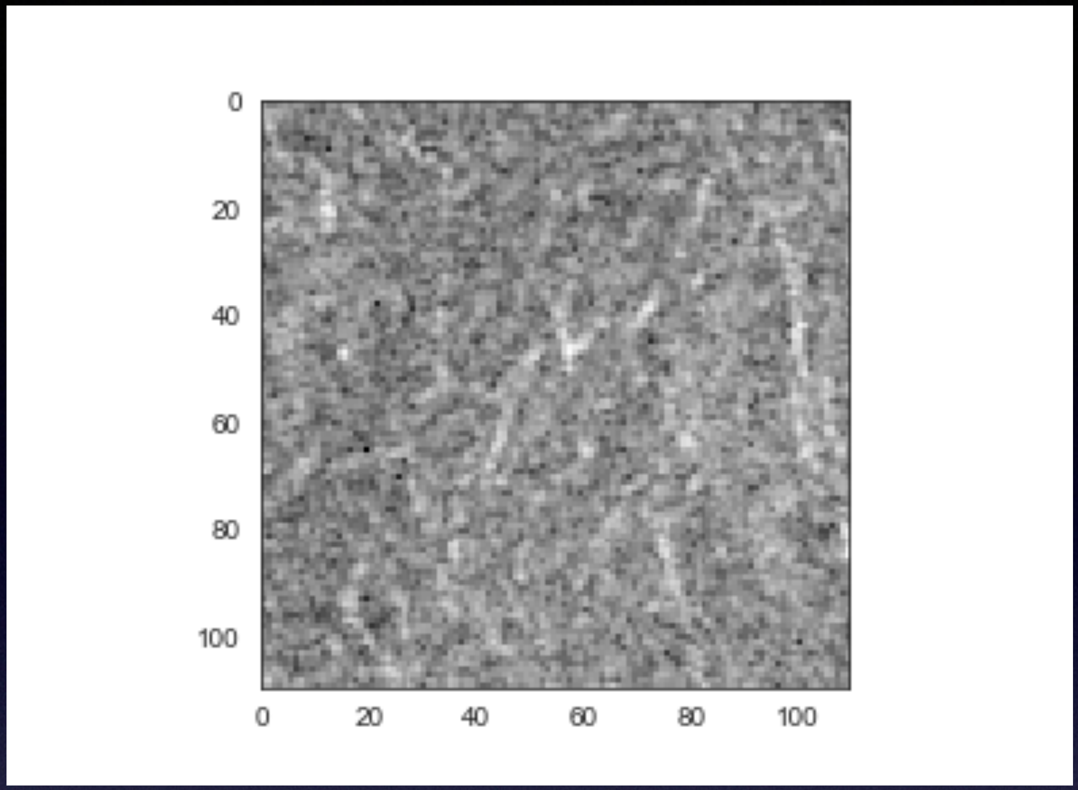


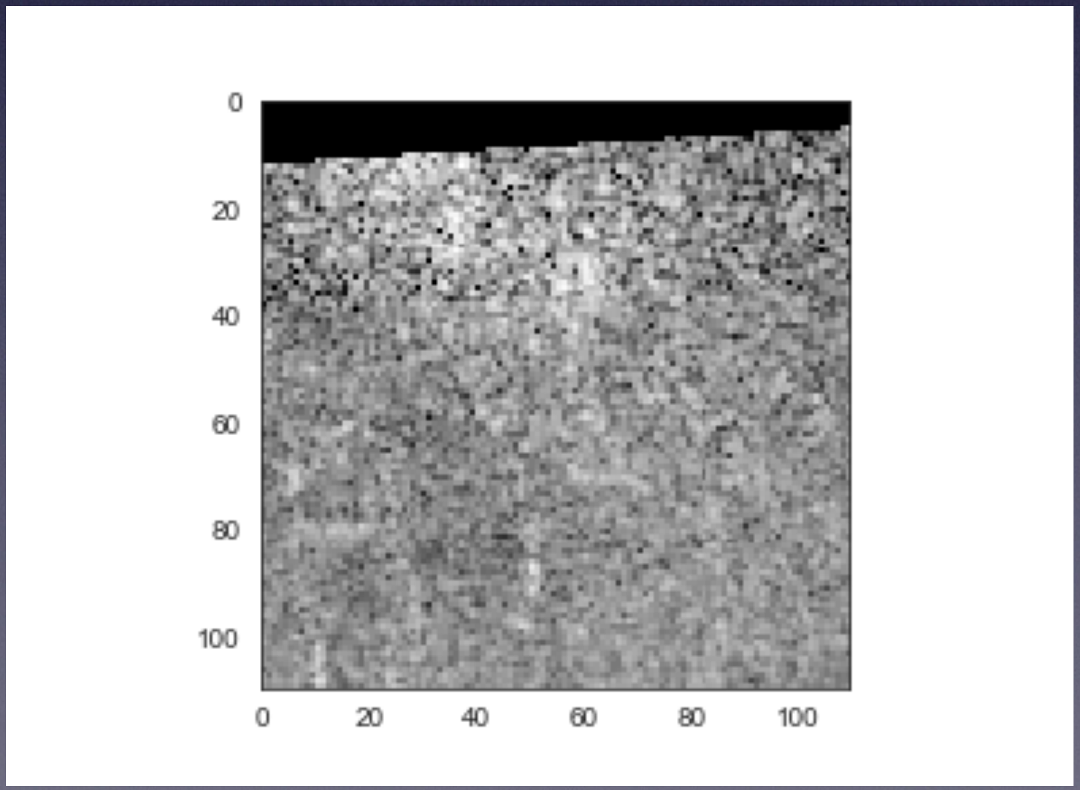


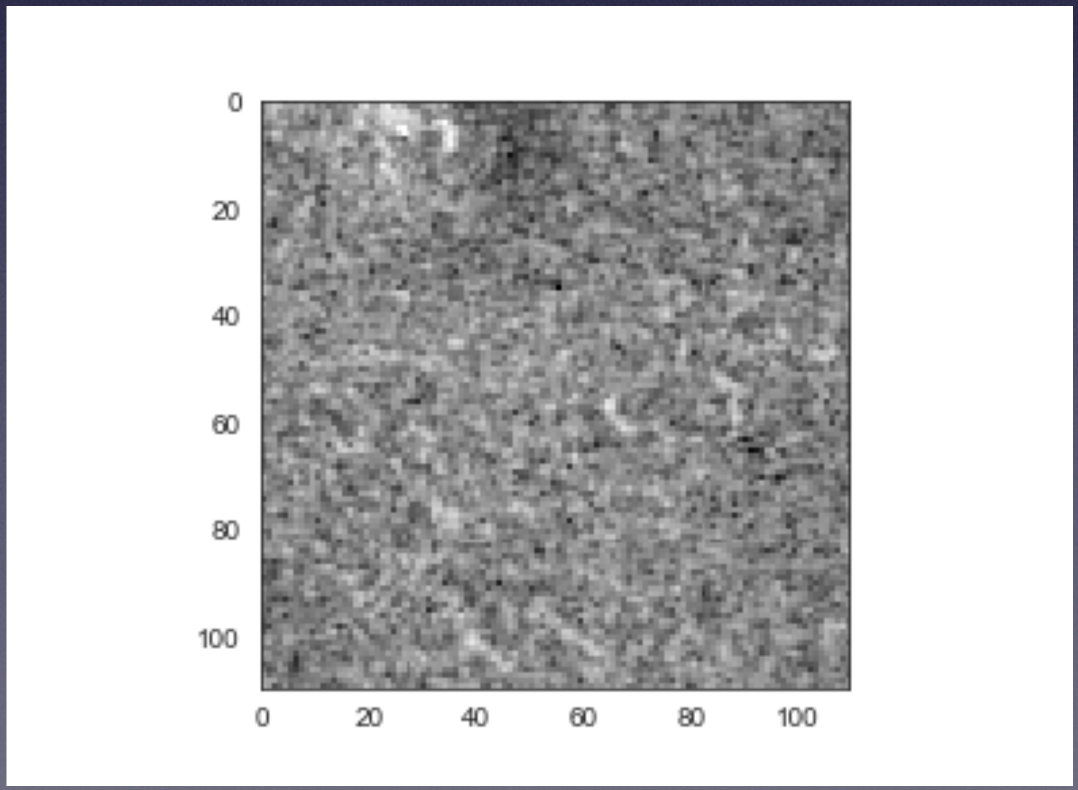


No space volcanos :(



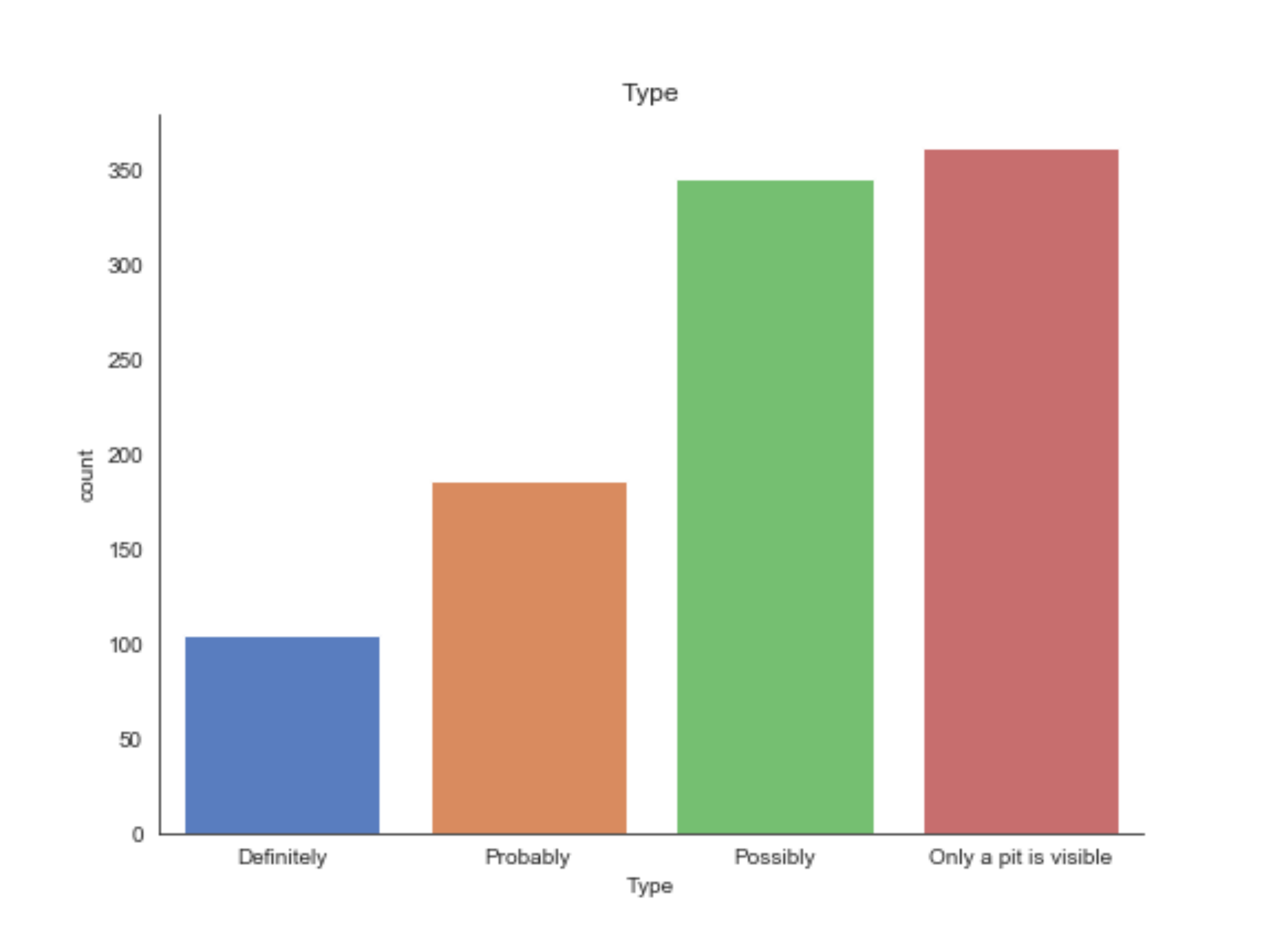






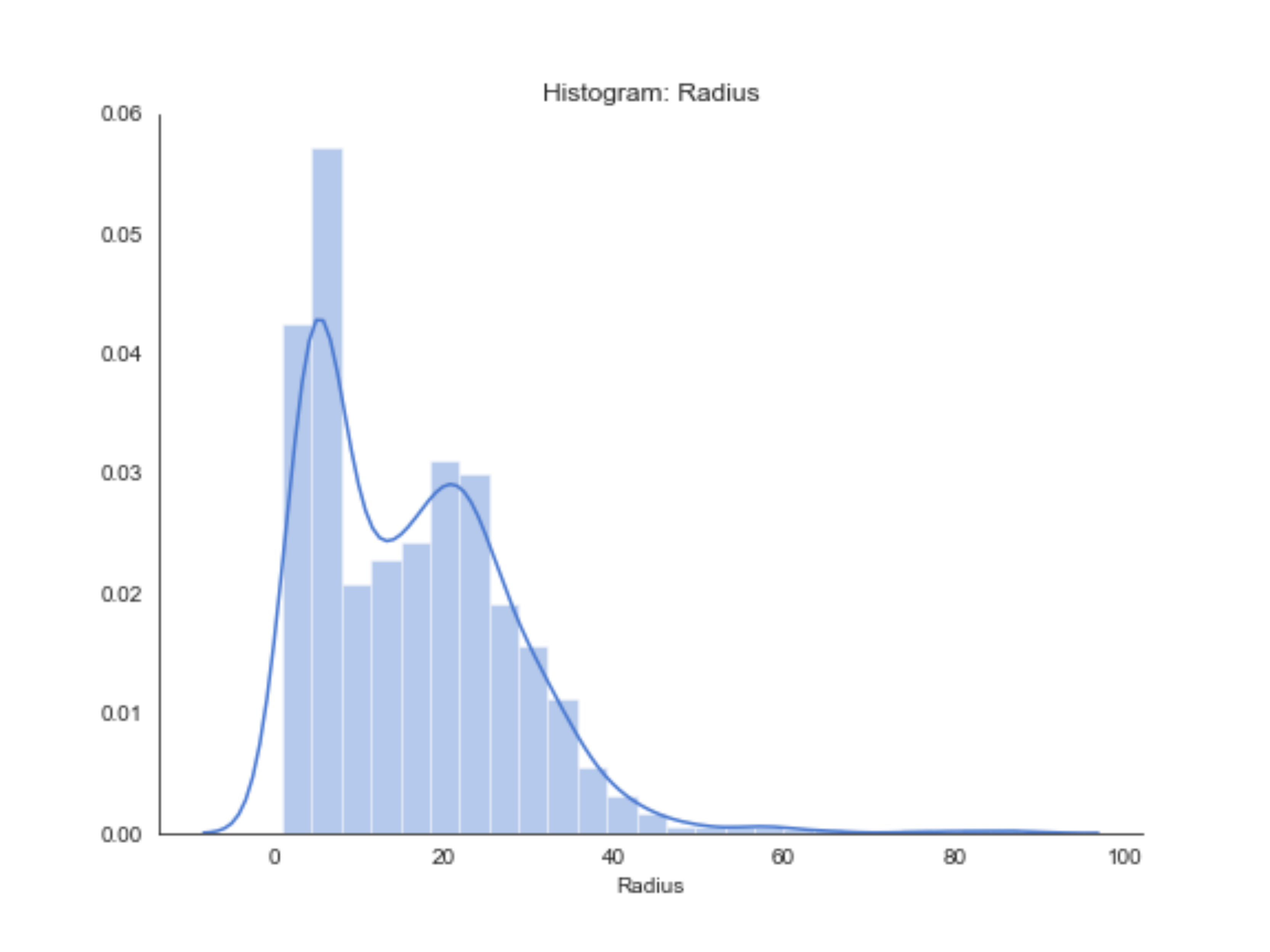
Additional variables in the dataset

Type: describes the degree of certainty of a positive classification



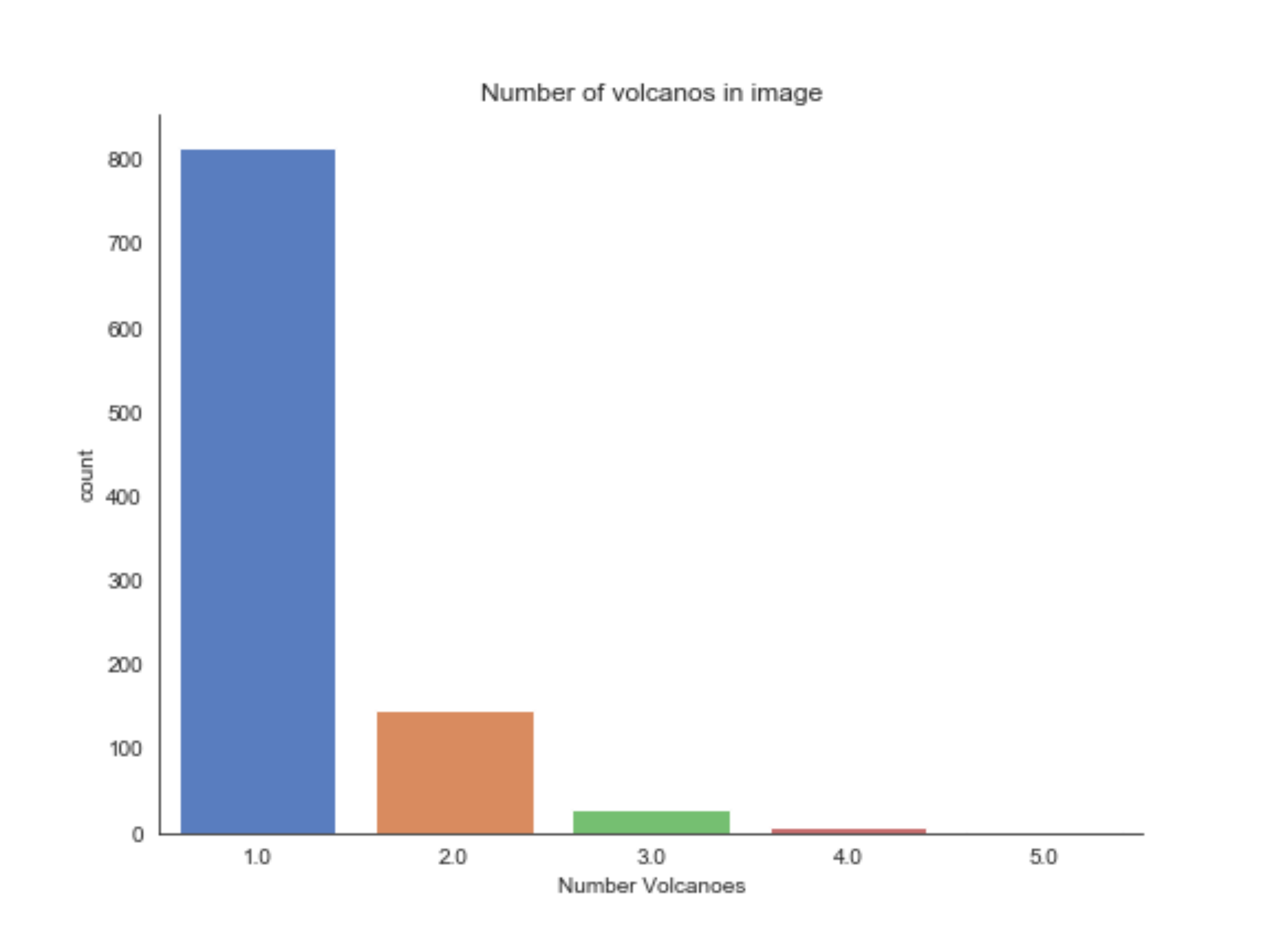
Additional variables in the dataset

Radius: measured in pixels



Additional variables in the dataset

Number of volcanoes in the image

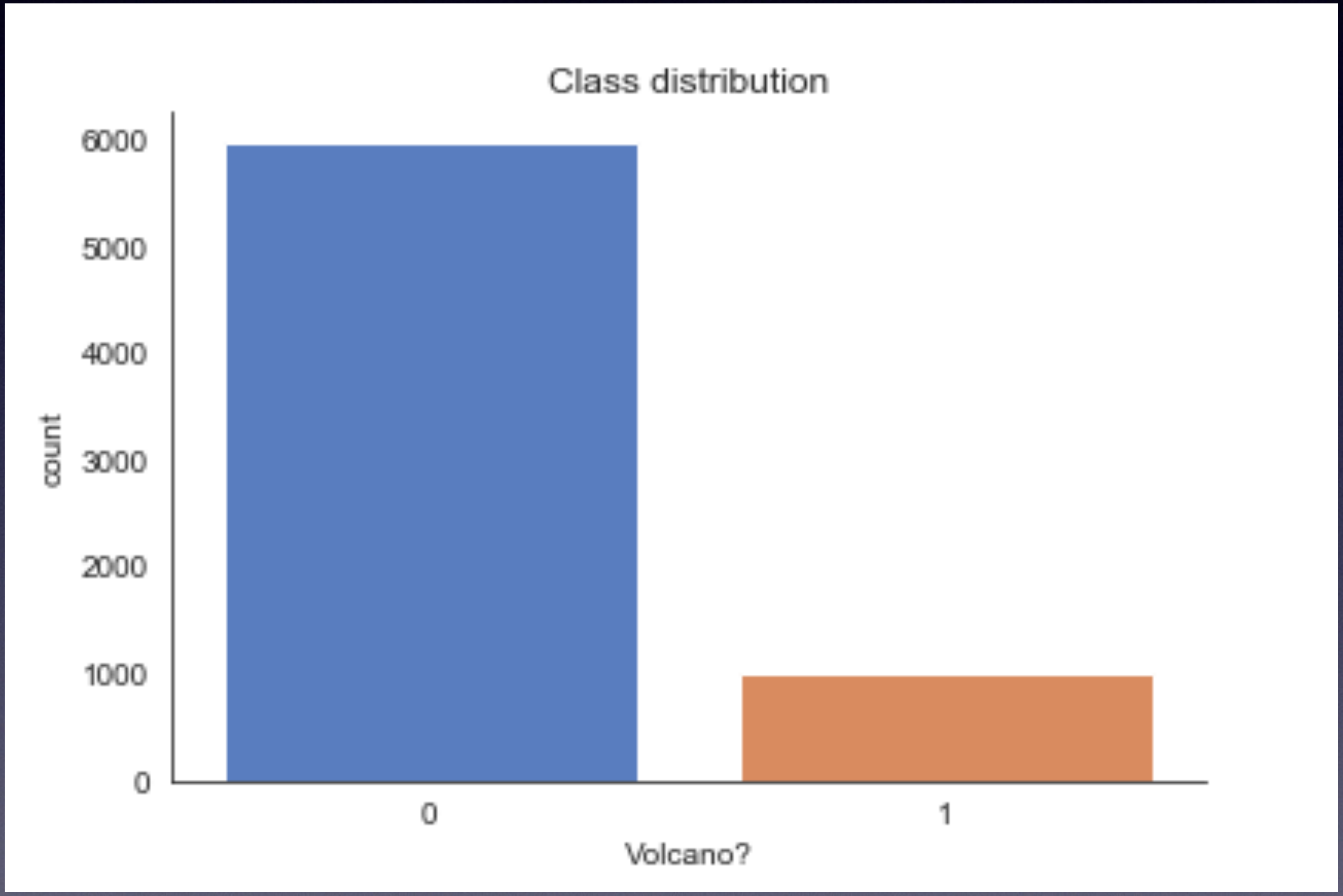


Class variable

Unbalanced class distribution

85% of the images do not contain volcanos

Used upsampling of the minority class to achieve 50% split of the training data



Evaluate algorithms

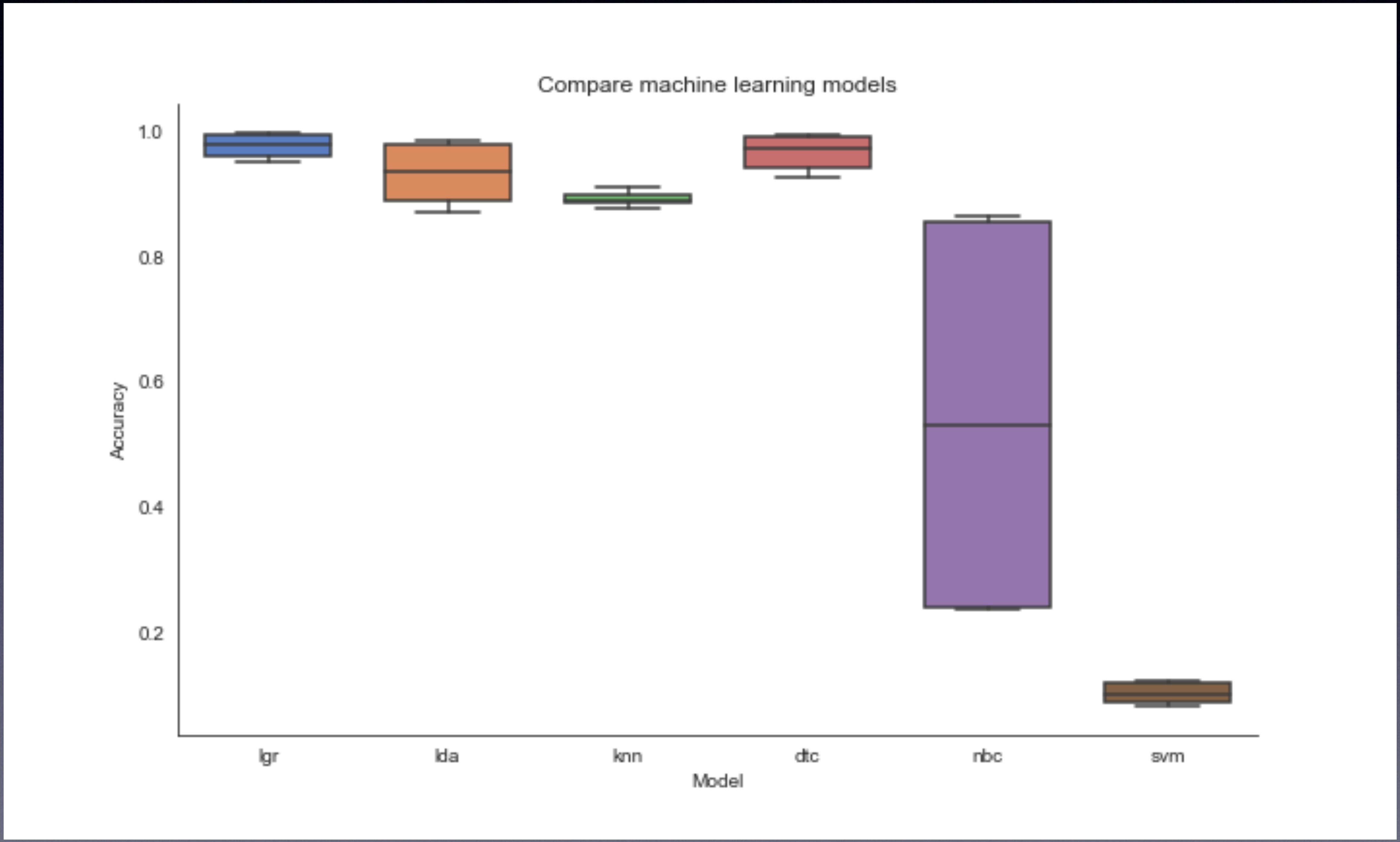
Spot check of machine learning algorithms using cross-validation (10 folds):

- Logistic regression
- Linear discriminant analysis
- K-Nearest neighbors
- Decision tree classifier
- Naive Bayes classifier
- Support vector classifier

Evaluate algorithms

	model	mean_train_acc	std_train_acc	test_acc	precision	recall	f1_score
0	lgr	0.975917	0.019305	0.930505	0.809645	0.735023	0.770531
3	dtc	0.966167	0.027361	0.884784	0.678679	0.520737	0.589309
1	lda	0.932583	0.047561	0.866862	0.572314	0.638249	0.603486
2	knn	0.892583	0.009534	0.820410	0.413897	0.315668	0.358170
5	svm	0.106083	0.016385	0.668252	0.294169	0.778802	0.427037
4	nbc	0.546000	0.302359	0.332846	0.173402	0.850230	0.288056

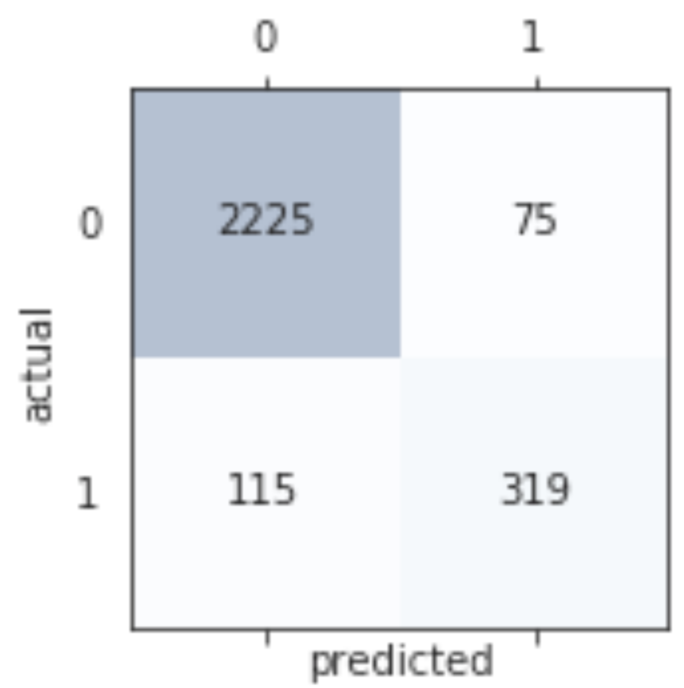
Evaluate algorithms



Evaluate algorithms

Wow…

```
###### Results ######
Model: lgr
Mean Training Accuracy: 0.976
Std Training Accuracy: 0.019
Test Accuracy: 0.931
Precision: 0.810
Recall: 0.735
F1 Score: 0.771
```



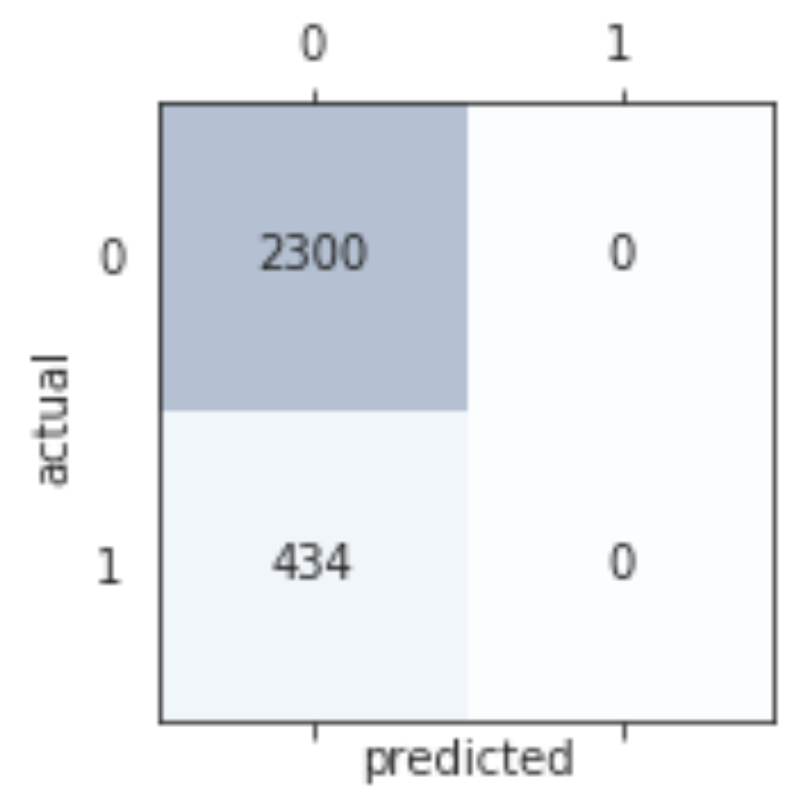
Evaluate algorithms

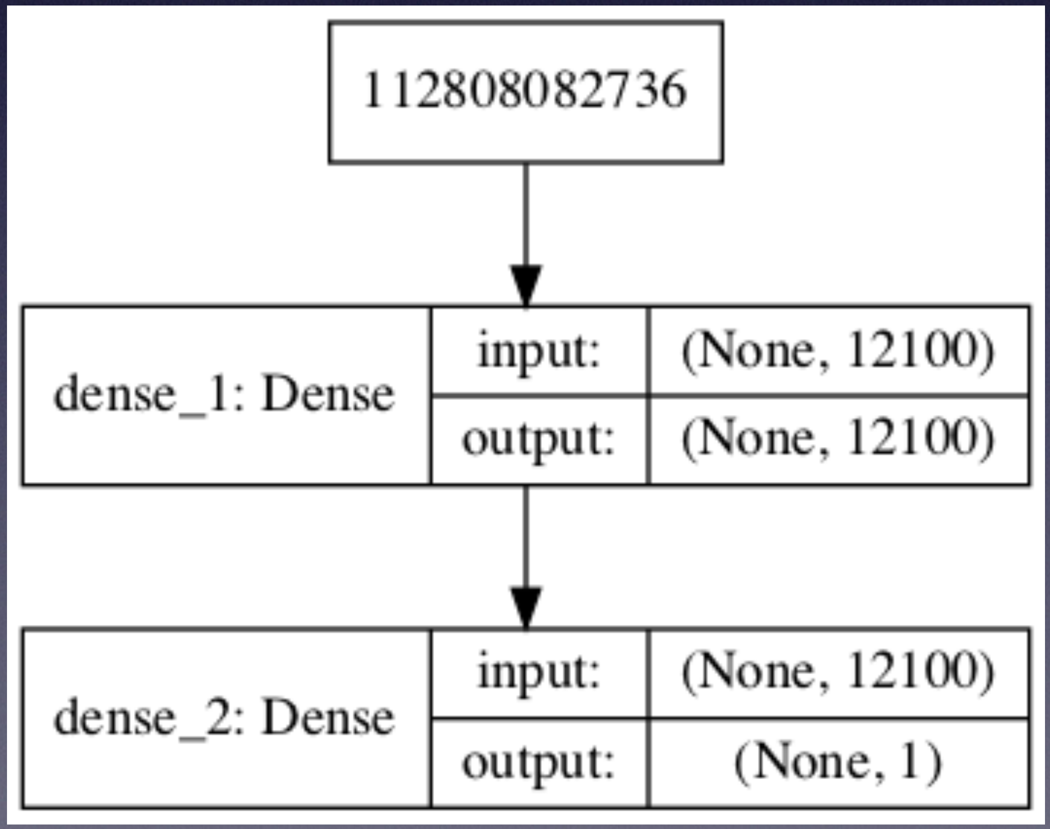
Five neural network models tested:

- One baseline neural network
- Four convolutional neural networks

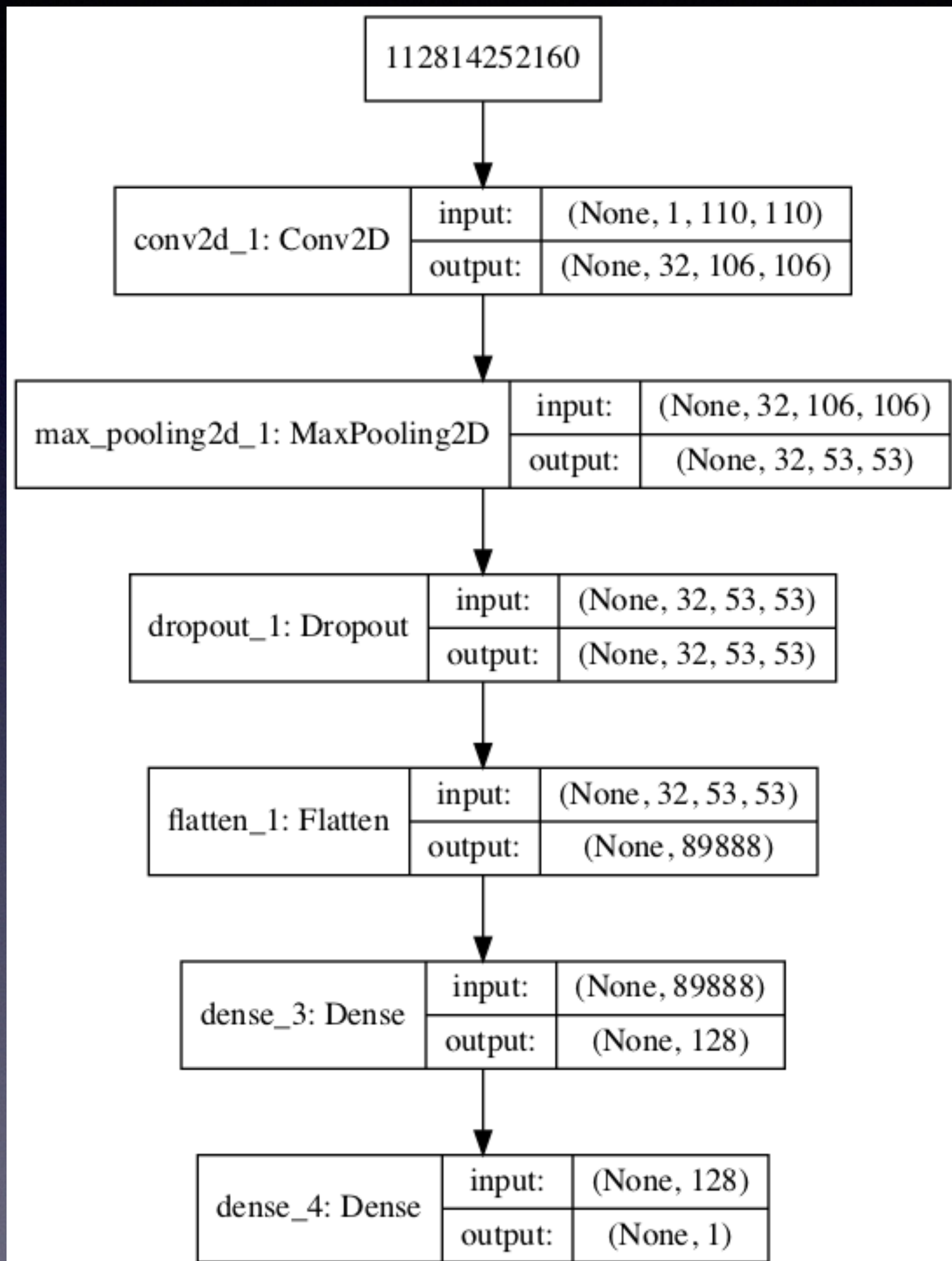
Baseline neural network model

```
###### Results ######
Model: nn_model
Training Accuracy: 0.500
Test Accuracy: 0.841
Precision: 0.000
Recall: 0.000
F1 Score: 0.000
```

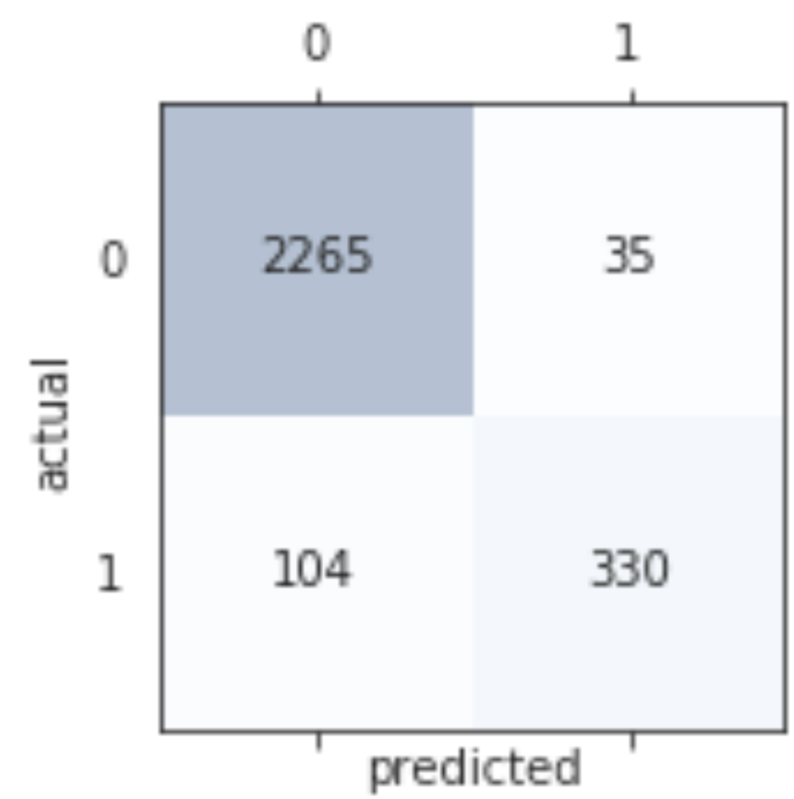




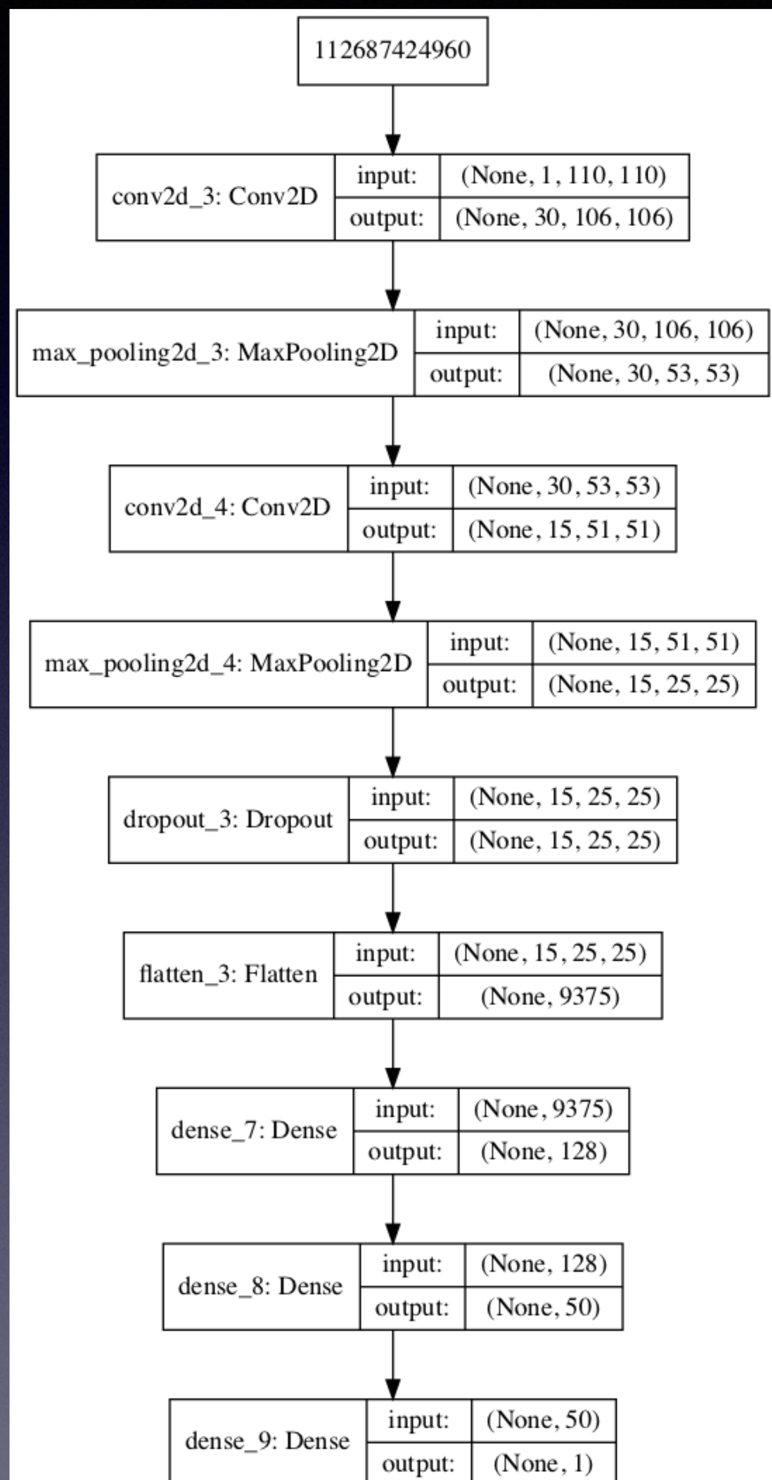
Convolutional neural network model # 1



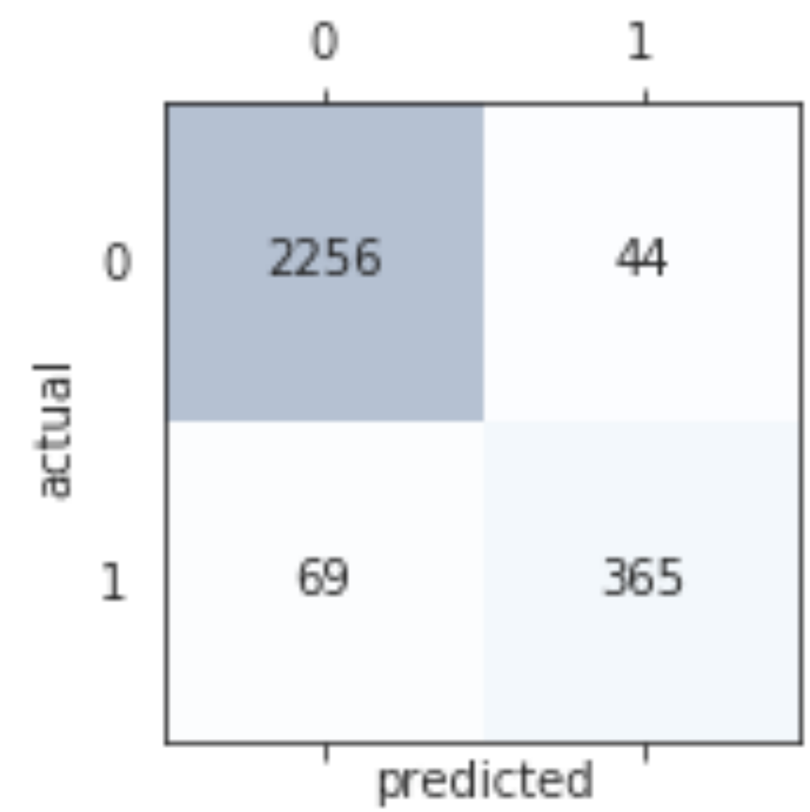

```
###### Results ######
Model: cnn_1_model
Training Accuracy: 1.000
Test Accuracy: 0.949
Precision: 0.904
Recall: 0.760
F1 Score: 0.826
```



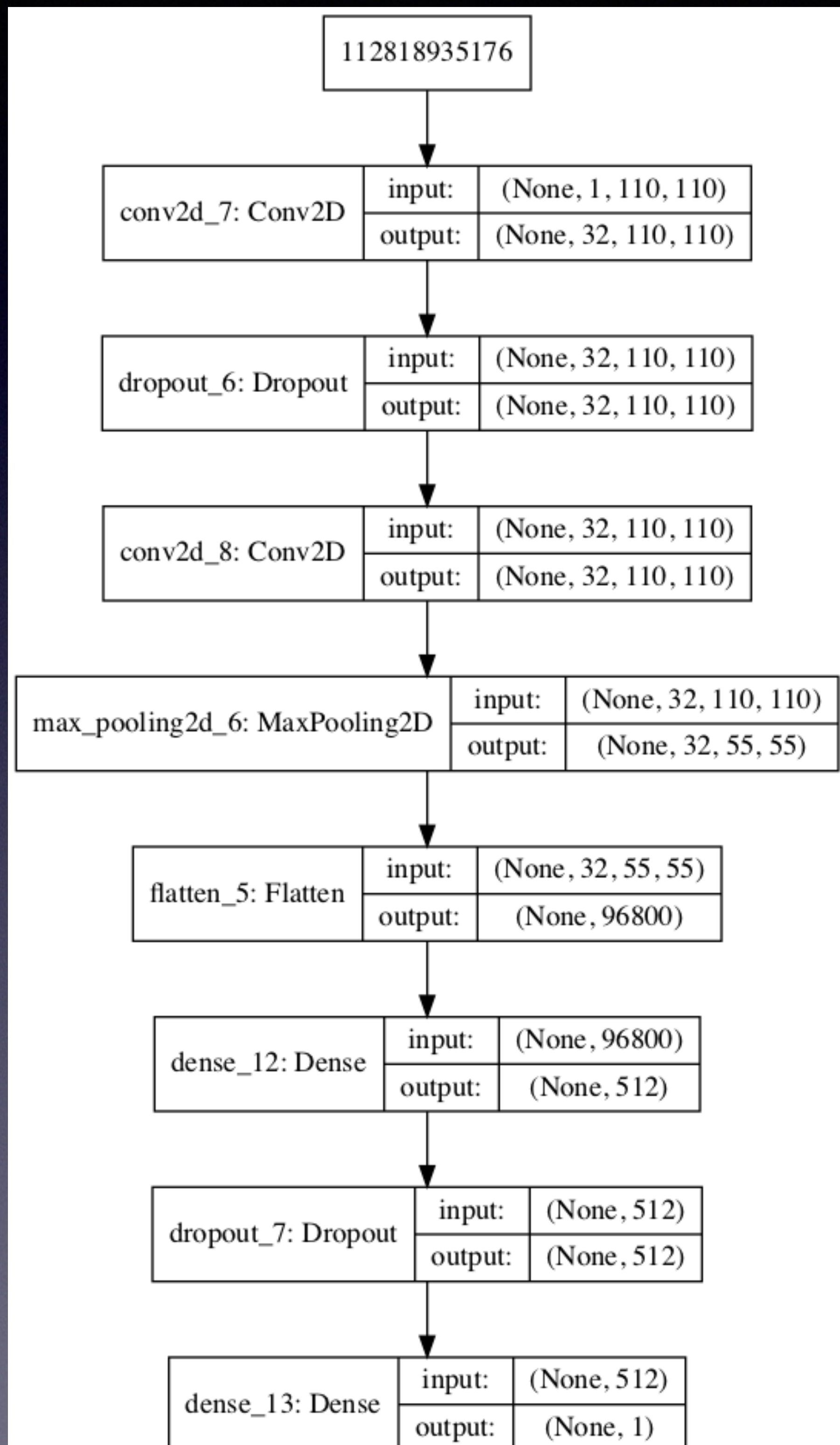
Convolutional neural network model # 2



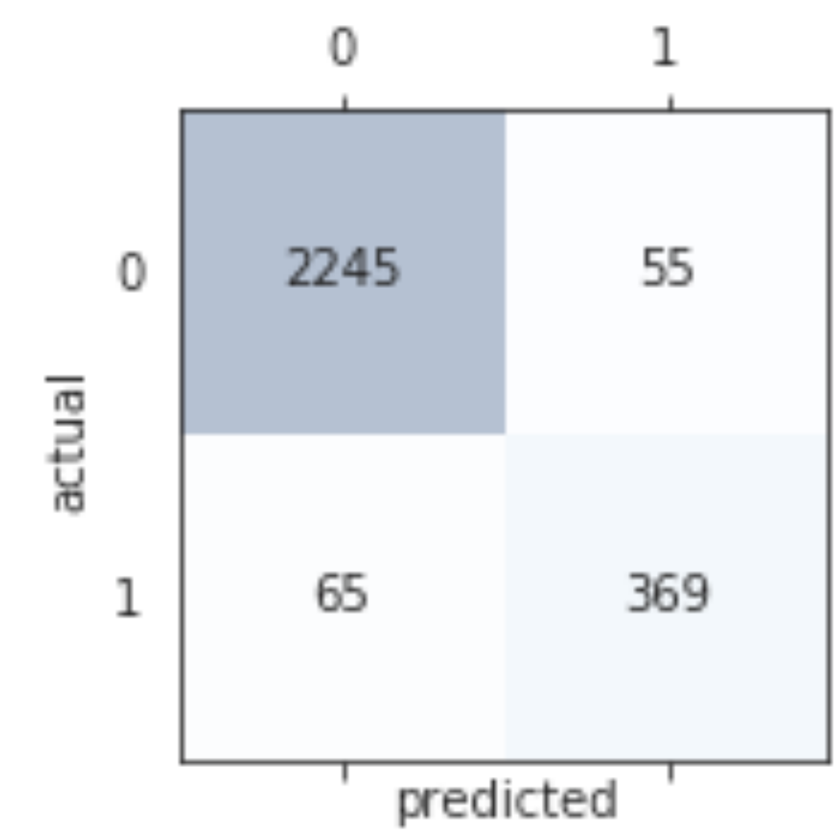

```
###### Results ######
Model: cnn_2_model
Training Accuracy: 1.000
Test Accuracy: 0.959
Precision: 0.892
Recall: 0.841
F1 Score: 0.866
```



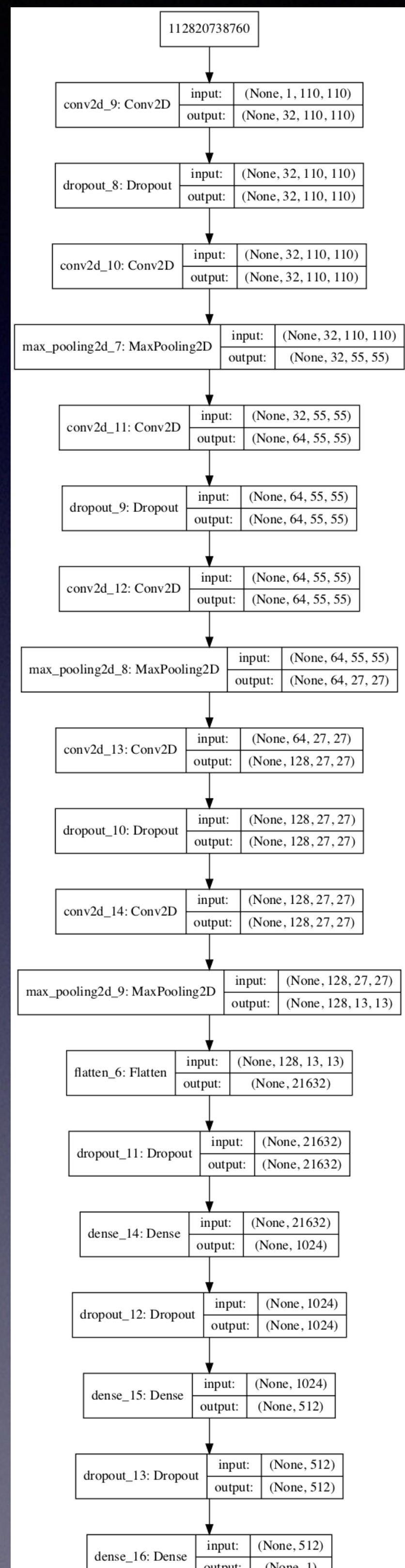
Convolutional neural network model # 3



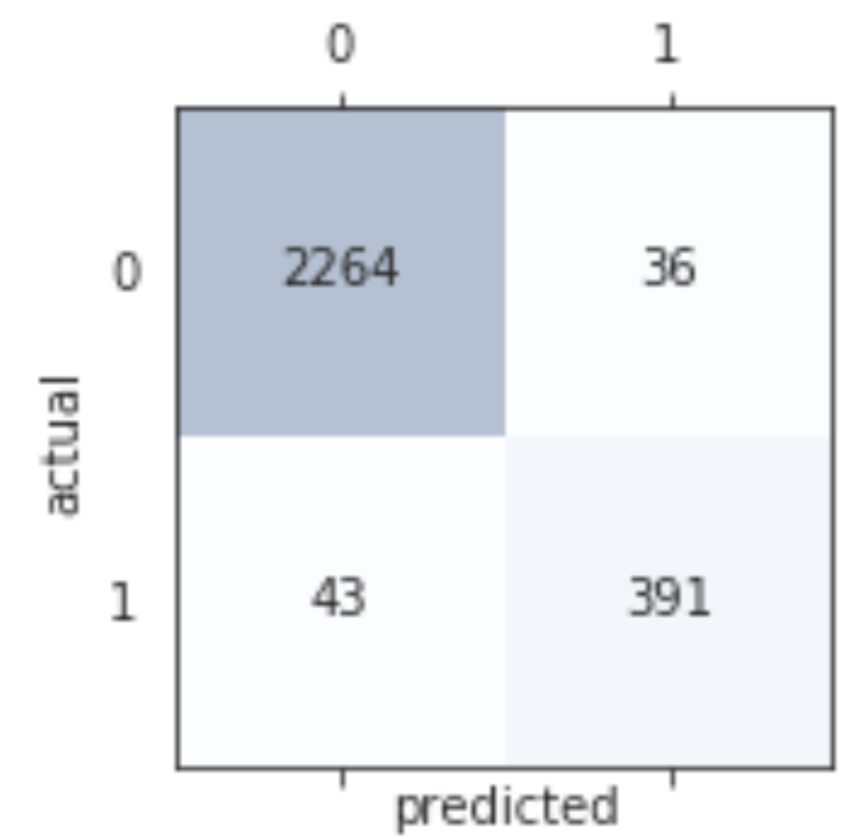
Results
Model: cnn_3_model
Training Accuracy: 0.997
Test Accuracy: 0.956
Precision: 0.870
Recall: 0.850
F1 Score: 0.860



Convolutional neural network model # 4




```
###### Results ######
Model: cnn_4_model
Training Accuracy: 0.999
Test Accuracy: 0.971
Precision: 0.916
Recall: 0.901
F1 Score: 0.908
```



Evaluate algorithms

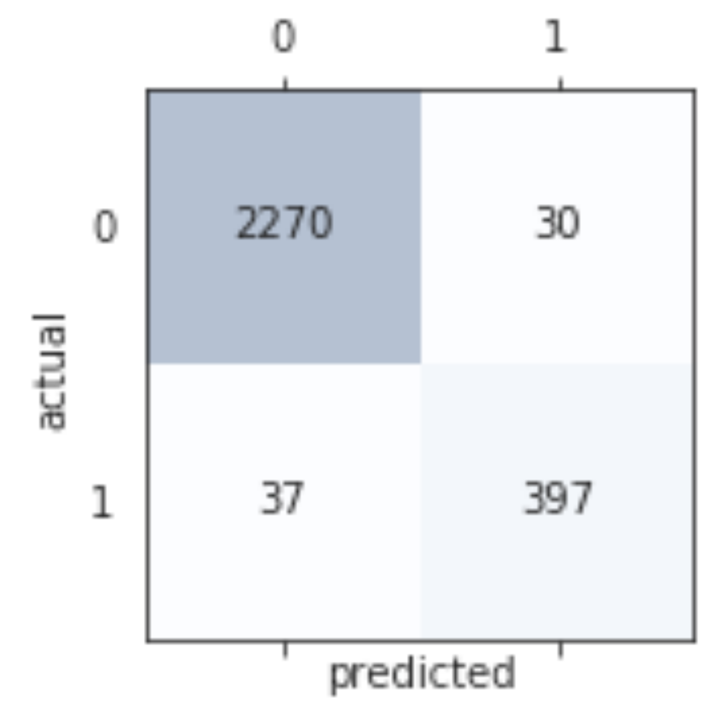
	model	train_acc	test_acc	precision	recall	f1_score
0	cnn_4_model	0.998583	0.971105	0.915691	0.900922	0.908246
0	cnn_2_model	0.999833	0.958669	0.892421	0.841014	0.865955
0	cnn_3_model	0.996917	0.956108	0.870283	0.850230	0.860140
0	cnn_1_model	1.000000	0.949159	0.904110	0.760369	0.826033
0	nn_model	0.500000	0.841258	0.000000	0.000000	0.000000

Finalize model

Trained CNN-4 model on the entire training dataset

Test-set accuracy: **97.5%**

```
###### Results ######
Model: cnn_4
Training Accuracy: 1.000
Test Accuracy: 0.975
Precision: 0.930
Recall: 0.915
F1 Score: 0.922
```



Finalize model

Opportunities for further improvement:

- Image augmentation techniques:
  - Random-shifts
  - Random-rotations
  - Random-flips
- Transfer learning (ResNet50 for example)