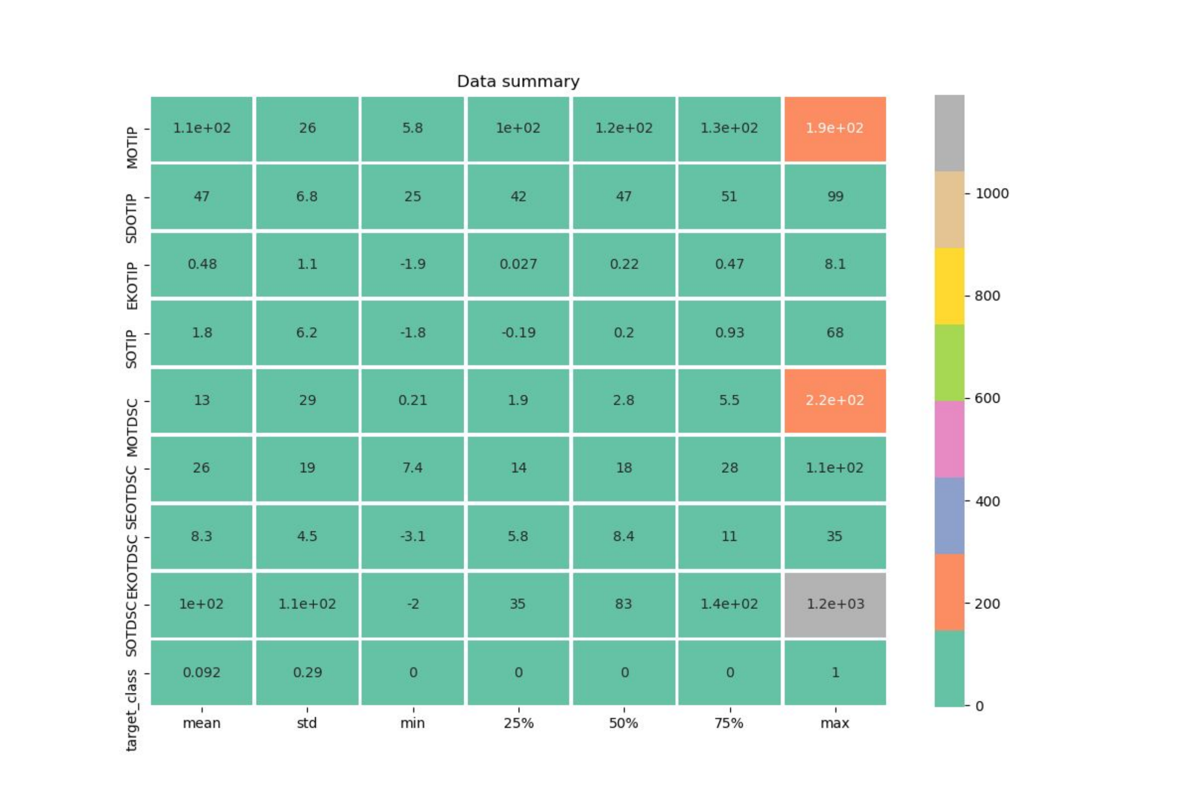
Pulsars prediction

1. Checking the dataset

We will need to scale it.

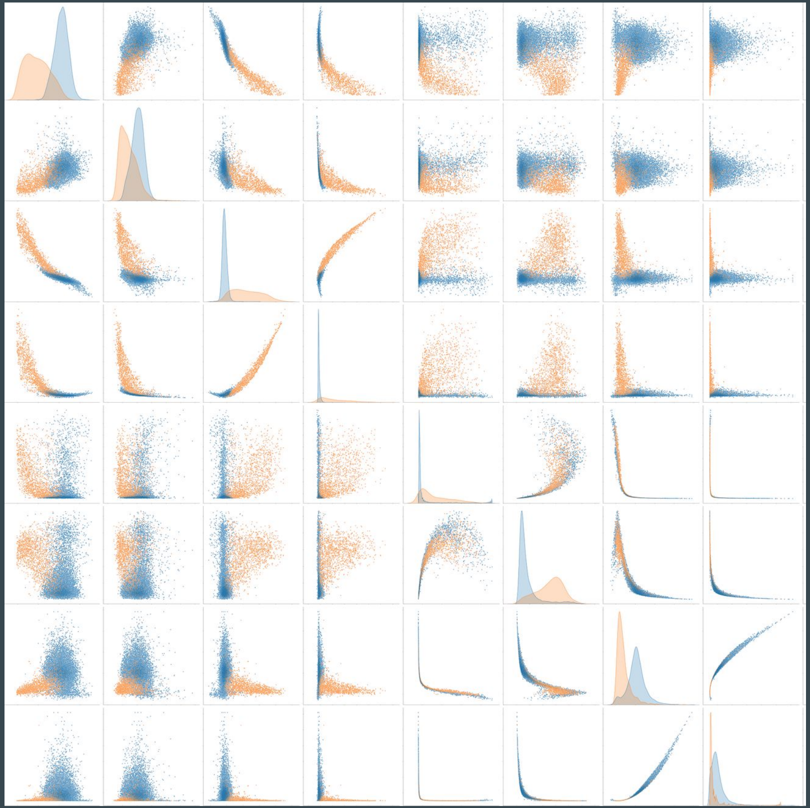
Using **StandardScaler**



Data summary

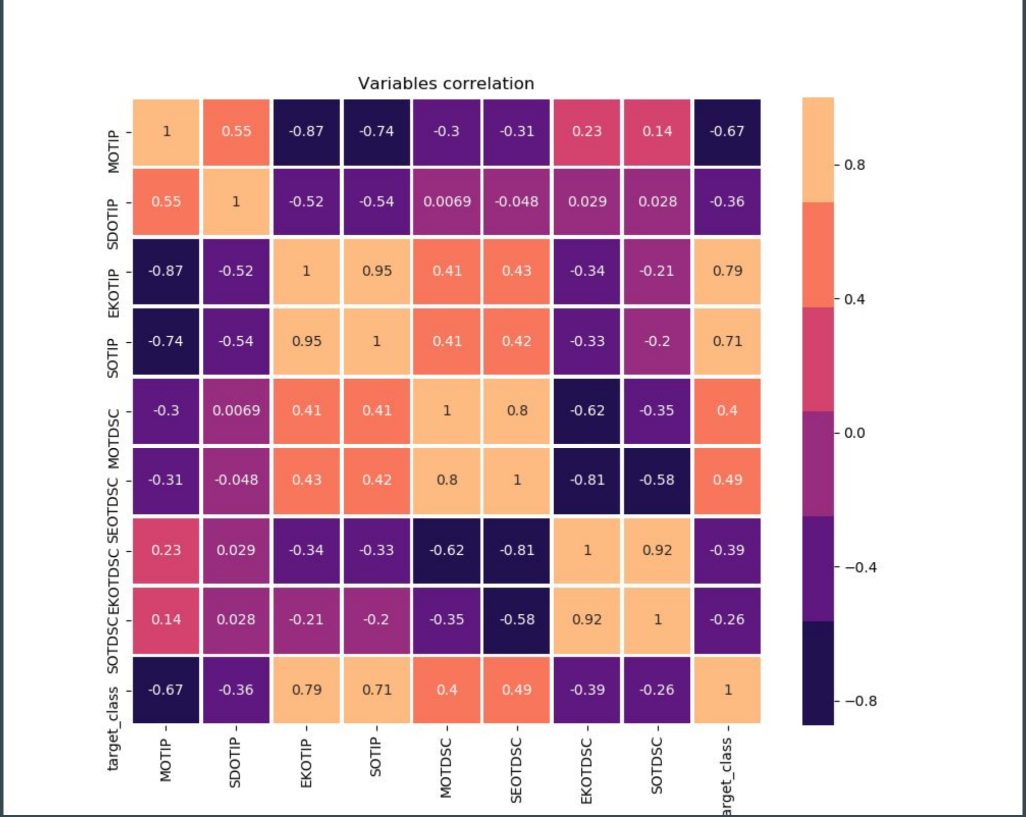
	mean	std	min	25%	50%	75%	max
MOTIP	1.1e+02	26	5.8	1e+02	1.2e+02	1.3e+02	1.9e+02
SDOTIP	47	6.8	25	42	47	51	99
EKOTIP	0.48	1.1	-1.9	0.027	0.22	0.47	8.1
SOTIP	1.8	6.2	-1.8	-0.19	0.2	0.93	68
MOTDSC	13	29	0.21	1.9	2.8	5.5	2.2e+02
SEOTDSC	26	19	7.4	14	18	28	1.1e+02
EKOTDSC	8.3	4.5	-3.1	5.8	8.4	11	35
SOTDSC	1e+02	1.1e+02	-2	35	83	1.4e+02	1.2e+03
target_class	0.092	0.29	0	0	0	0	1

1. Checking the dataset



1. Checking the dataset

There is a strong correlation between **SOTIP** and **EKOTIP** and also between **SOTDSC** and **EKOTDSC**, so we will drop **SOTIP** and **SOTDSC**

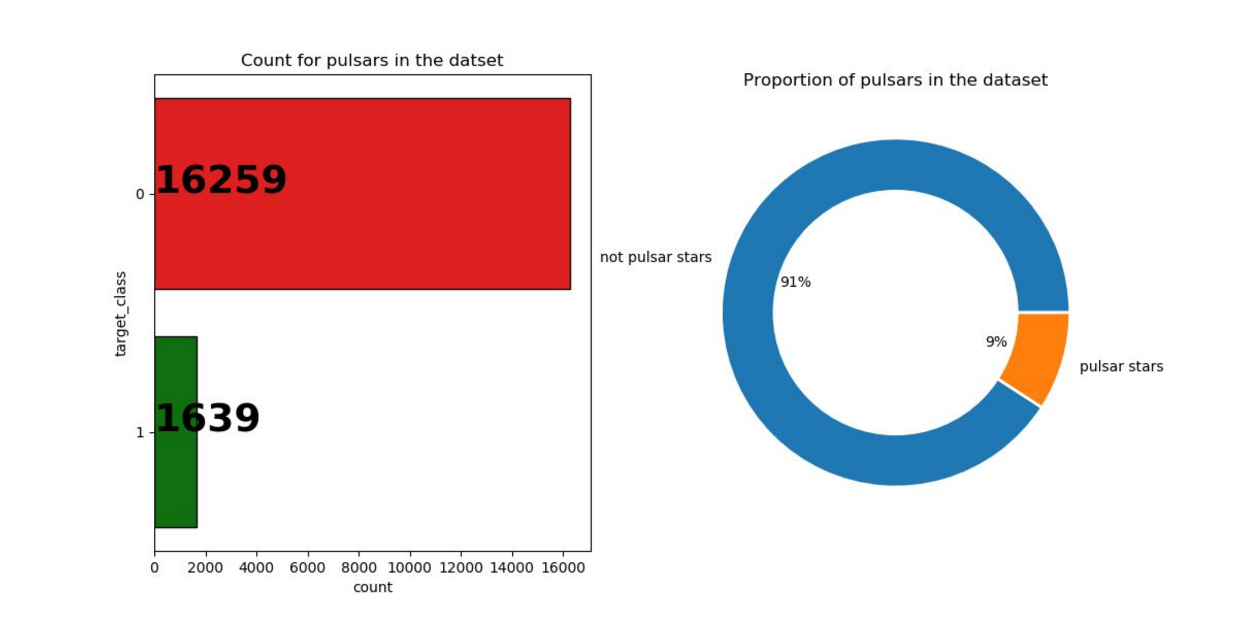


Variables correlation

	MOTIP	SDOTIP	EKOTIP	SOTIP	MOTDSC	SEOTDSC	EKOTDSC	SOTDSC	arget_class
MOTIP	1	0.55	-0.87	-0.74	-0.3	-0.31	0.23	0.14	-0.67
SDOTIP	0.55	1	-0.52	-0.54	0.0069	-0.048	0.029	0.028	-0.36
EKOTIP	-0.87	-0.52	1	0.95	0.41	0.43	-0.34	-0.21	0.79
SOTIP	-0.74	-0.54	0.95	1	0.41	0.42	-0.33	-0.2	0.71
MOTDSC	-0.3	0.0069	0.41	0.41	1	0.8	-0.62	-0.35	0.4
SEOTDSC	-0.31	-0.048	0.43	0.42	0.8	1	-0.81	-0.58	0.49
EKOTDSC	0.23	0.029	-0.34	-0.33	-0.62	-0.81	1	0.92	-0.39
SOTDSC	0.14	0.028	-0.21	-0.2	-0.35	-0.58	0.92	1	-0.26
target_class	-0.67	-0.36	0.79	0.71	0.4	0.49	-0.39	-0.26	1

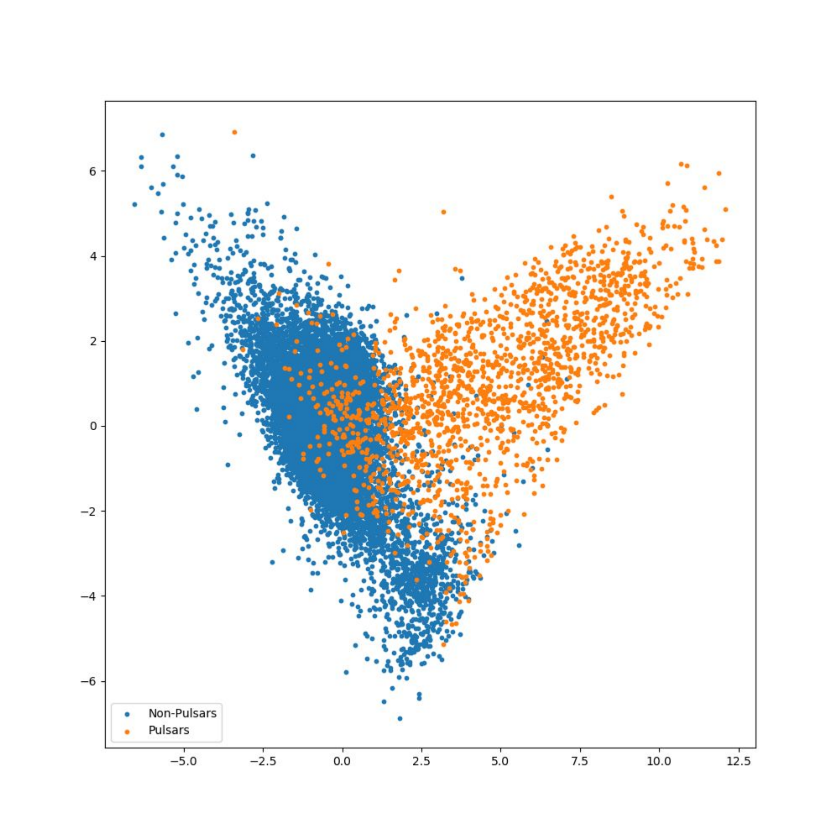
1. Checking the dataset

Dataset is imbalanced - we will need to use stratification



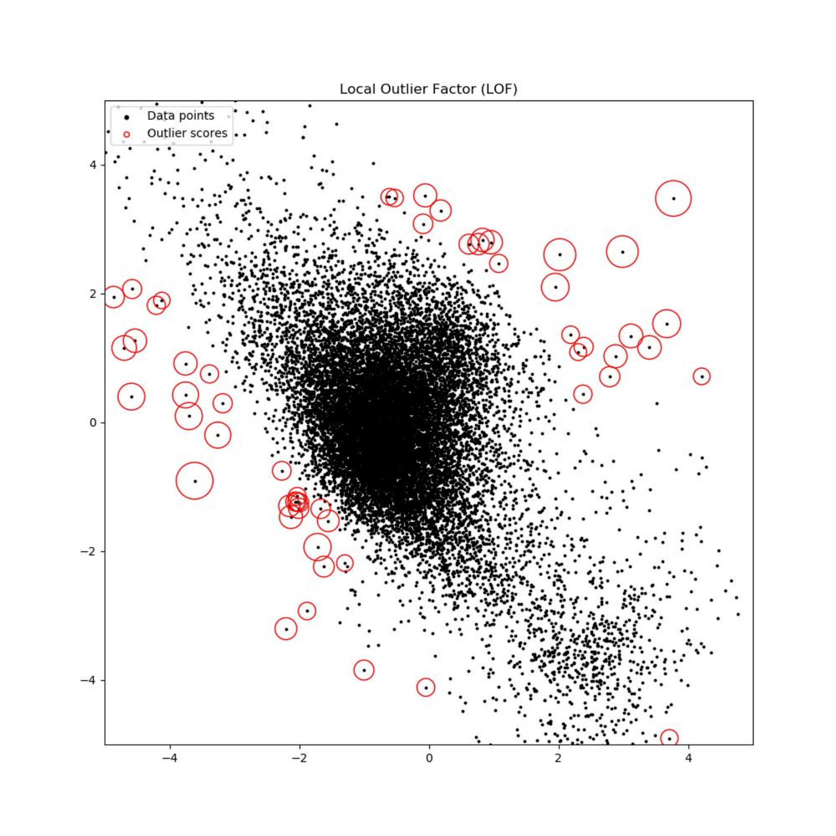
1. Checking the dataset

Let's search for anomalies using Principal Component Analysis and Local Outlier Factor



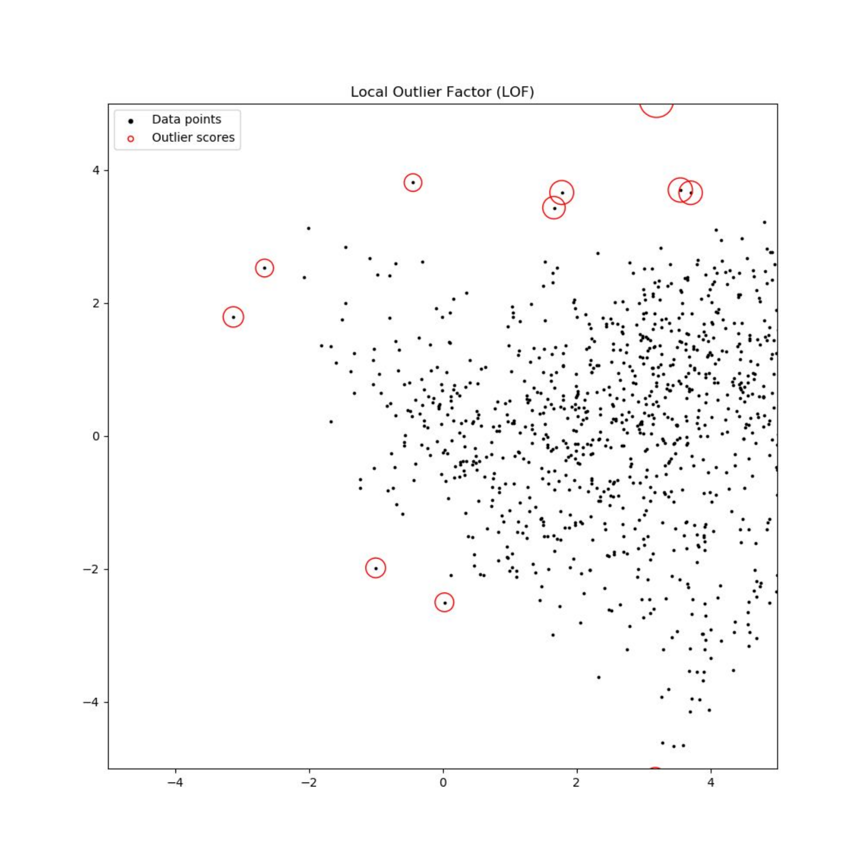
1. Checking the dataset

We can see some outliers for non-pulsars...



1. Checking the dataset

...and for pulsars too.
Let’s remove them.



1. Checking the dataset

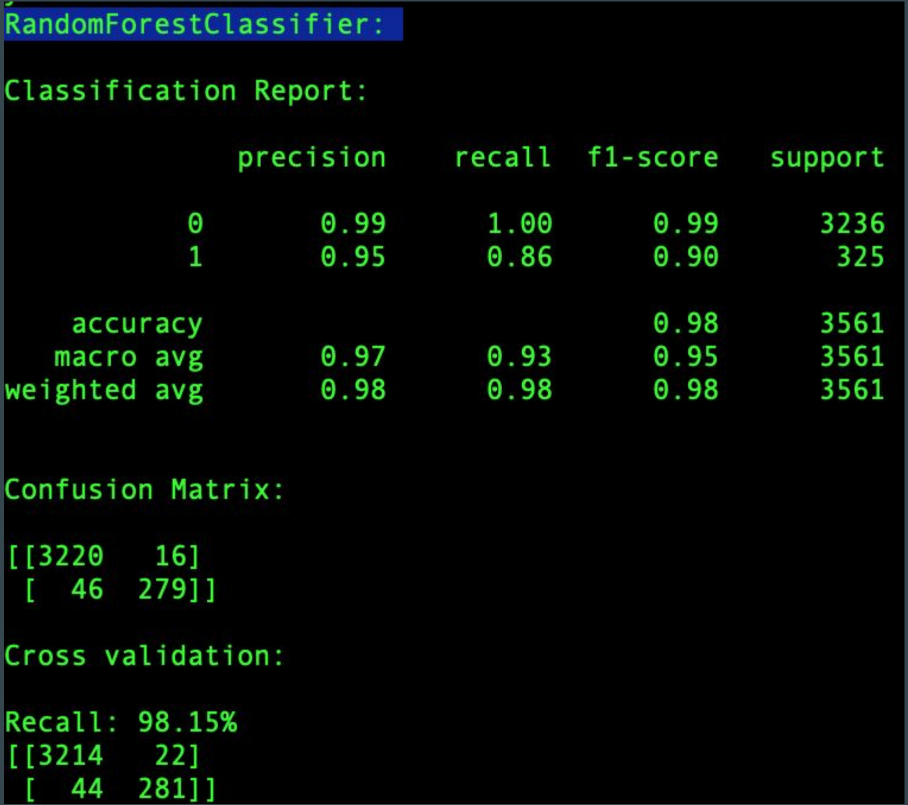
Splitting the dataset on train, test and validation

```
X_train, X_test, y_train, y_test = train_test_split(X, y,  test_size=0.2,
random_state=0, stratify=y)
```

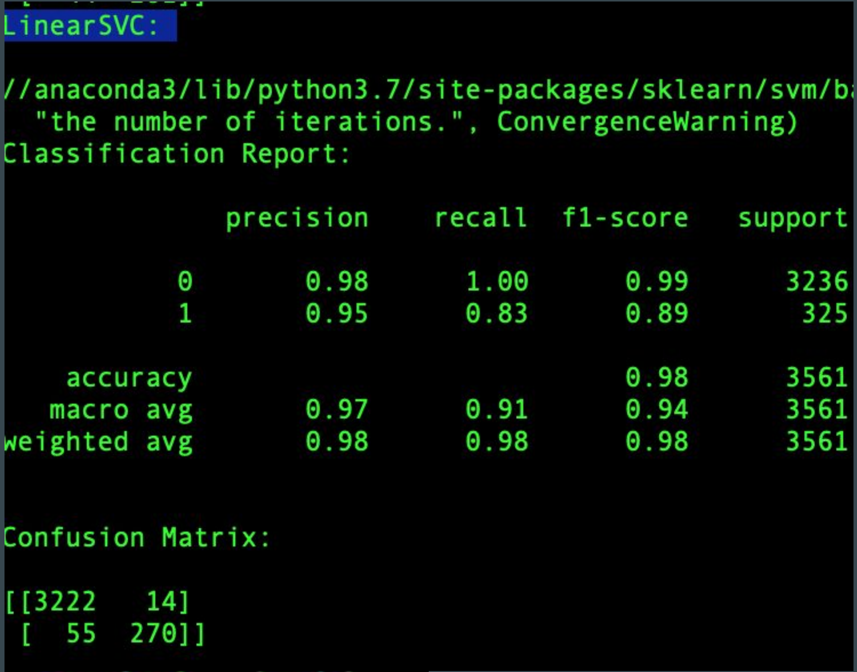
```
X_train, X_val, y_train, y_val = train_test_split(X_train, y_train,
test_size=0.1, random_state=0, stratify=y_train)
```

And then go on testing different models

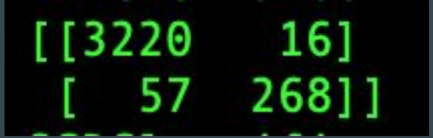
2. Testing models



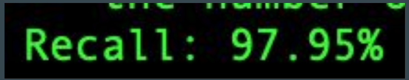
```
RandomForestClassifier:

Classification Report:

              precision    recall  f1-score   support

           0       0.99      1.00      0.99      3236
           1       0.95      0.86      0.90       325

    accuracy                           0.98      3561
   macro avg       0.97      0.93      0.95      3561
weighted avg       0.98      0.98      0.98      3561


Confusion Matrix:

[[3220   16]
 [  46  279]]

Cross validation:

Recall: 98.15%
[[3214   22]
 [  44  281]]
```



```
LinearSVC:

//anaconda3/lib/python3.7/site-packages/sklearn/svm/ba
  "the number of iterations.", ConvergenceWarning)
Classification Report:

              precision    recall  f1-score   support

           0       0.98      1.00      0.99      3236
           1       0.95      0.83      0.89       325

    accuracy                           0.98      3561
   macro avg       0.97      0.91      0.94      3561
weighted avg       0.98      0.98      0.98      3561


Confusion Matrix:

[[3222   14]
 [  55  270]]
```

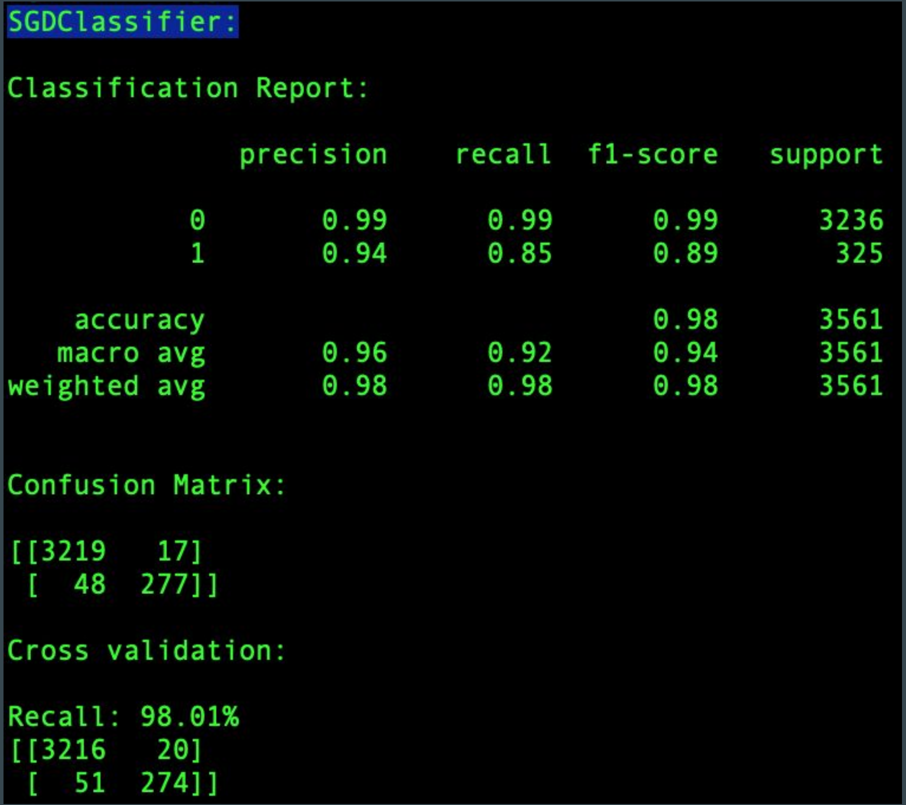


```
[[3220   16]
 [  57  268]]
```

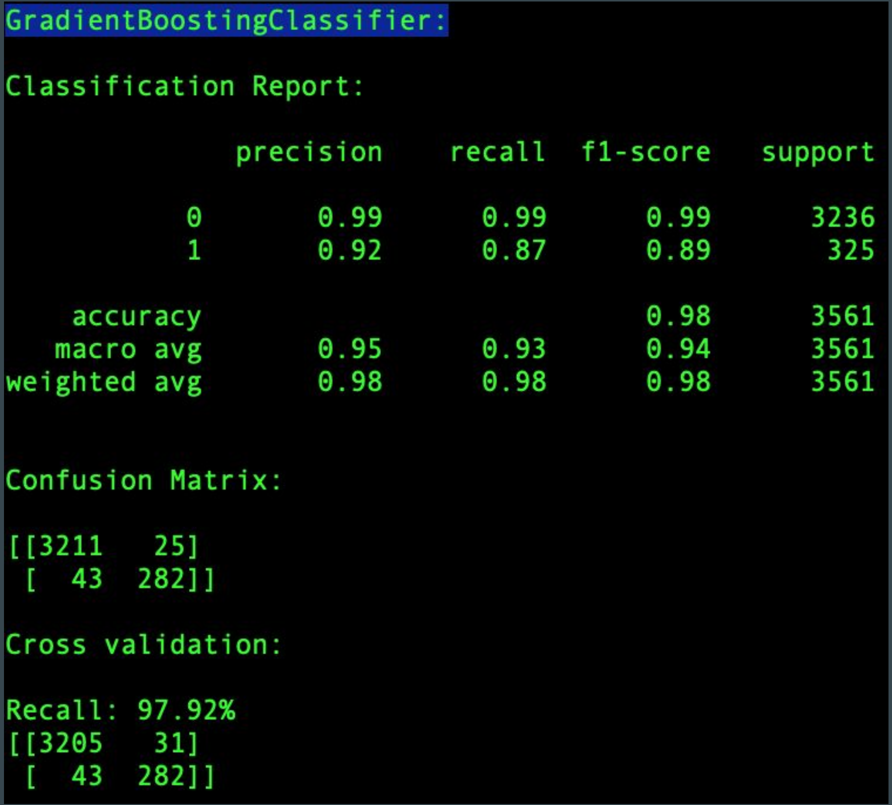


```
Recall: 97.95%
```

2. Testing models



```
SGDClassifier:

Classification Report:

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3236
           1       0.94      0.85      0.89       325

    accuracy                           0.98      3561
   macro avg       0.96      0.92      0.94      3561
weighted avg       0.98      0.98      0.98      3561


Confusion Matrix:

[[3219   17]
 [  48  277]]

Cross validation:

Recall: 98.01%
[[3216   20]
 [  51  274]]
```



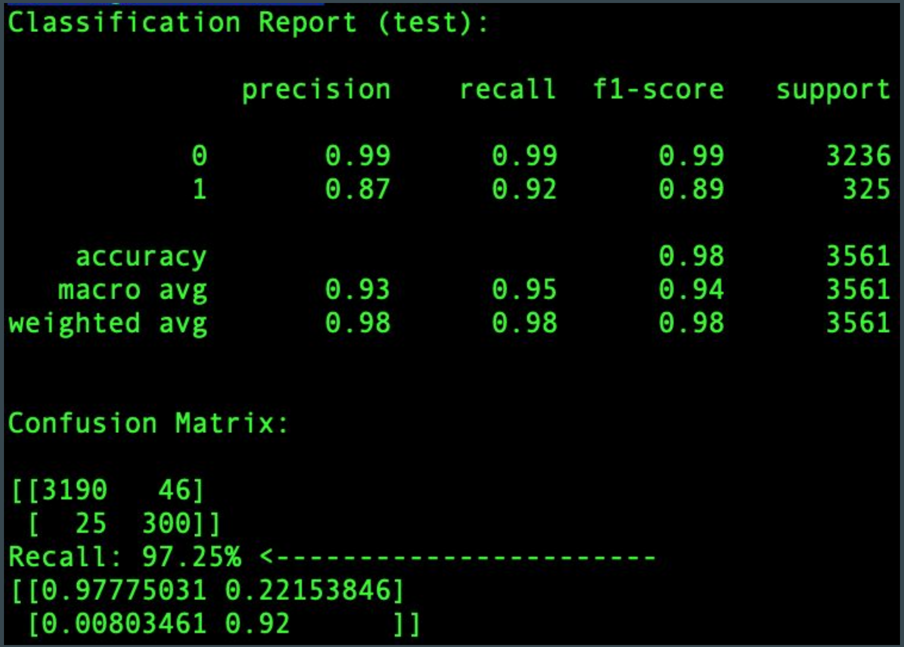
```
GradientBoostingClassifier:

Classification Report:

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3236
           1       0.92      0.87      0.89       325

    accuracy                           0.98      3561
   macro avg       0.95      0.93      0.94      3561
weighted avg       0.98      0.98      0.98      3561


Confusion Matrix:

[[3211   25]
 [  43  282]]

Cross validation:

Recall: 97.92%
[[3205   31]
 [  43  282]]
```

3. Tuning RF using **GridSearchCV**

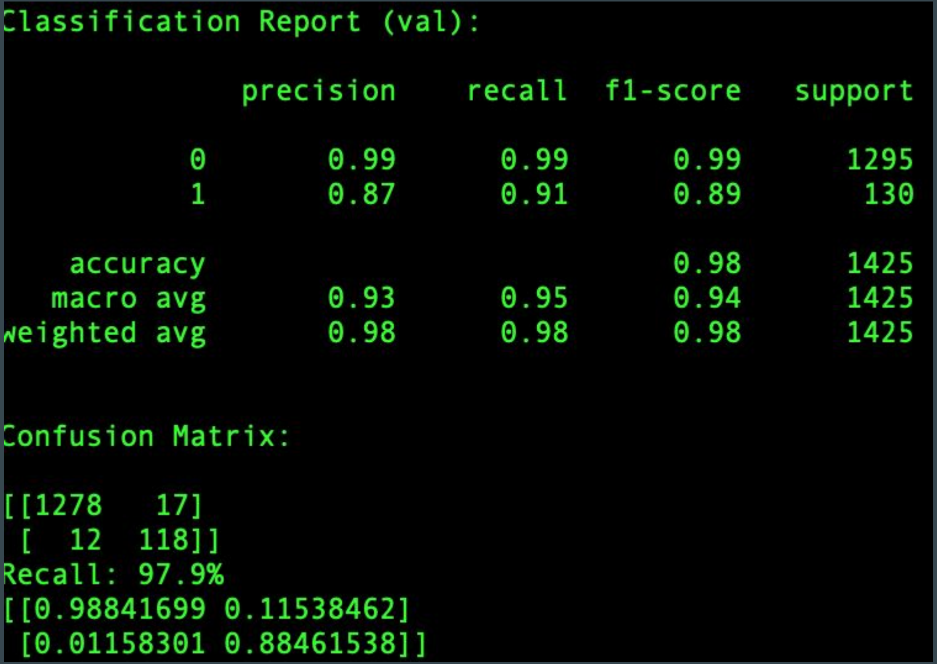
```
forest = RandomForestClassifier(bootstrap=True, class_weight='balanced_subsample',
                      criterion='gini', max_depth=15, max_features=4,
                      max_leaf_nodes=None, min_impurity_decrease=0.0,
                      min_impurity_split=None, min_samples_leaf=1,
                      min_samples_split=60, min_weight_fraction_leaf=0.0,
                      n_estimators=115, n_jobs=None, oob_score=False,
                      random_state=0, verbose=0, warm_start=False)
```

Plus, tune the model to miss as little pulsars as we can

3. Tuning RF using **GridSearchCV**

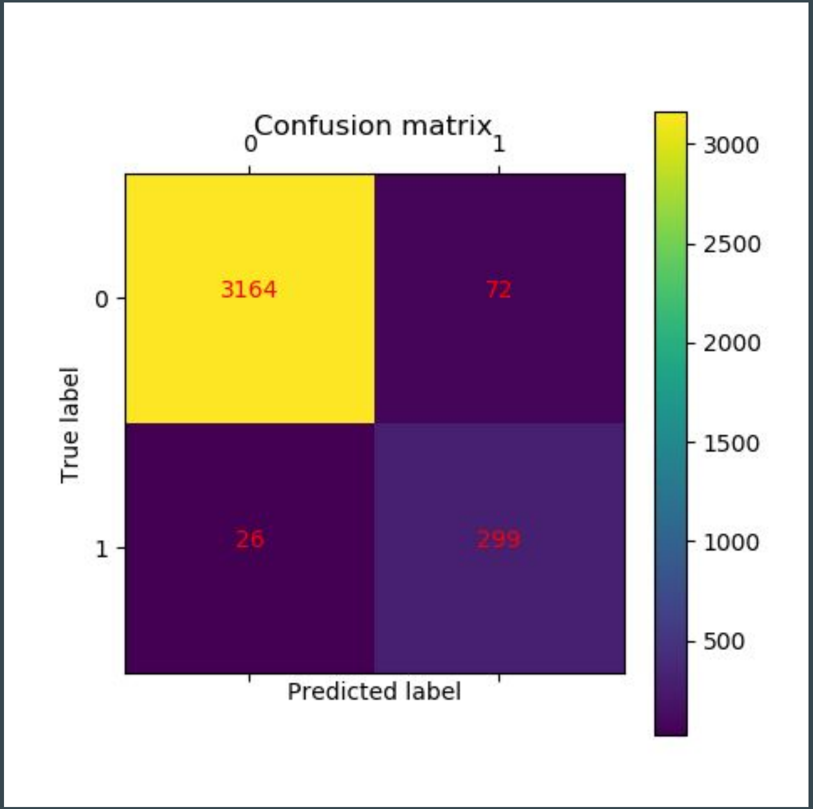


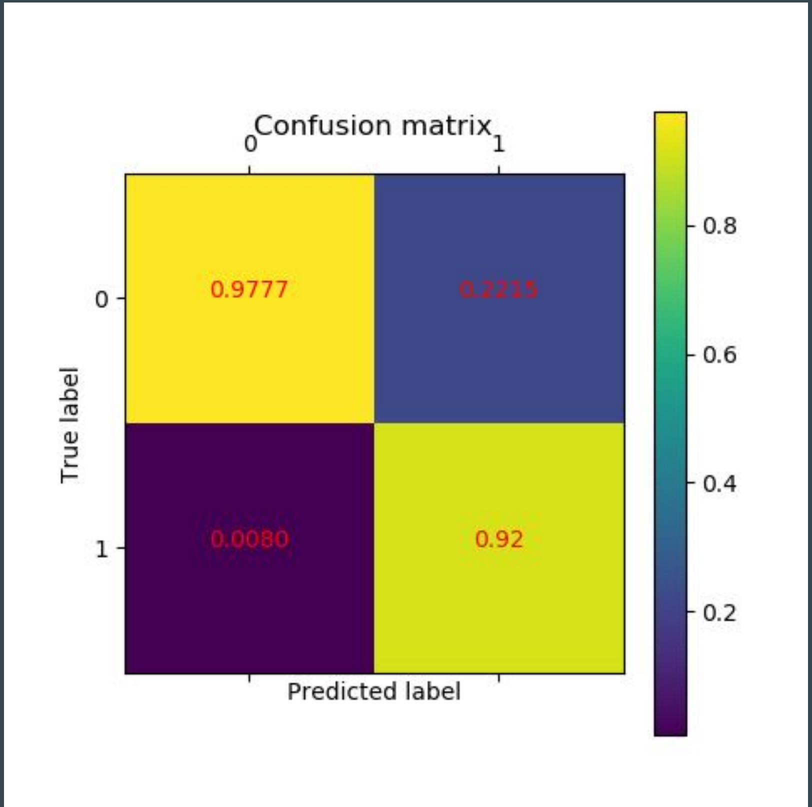
```
Classification Report (test):

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3236
           1       0.87      0.92      0.89       325

    accuracy                           0.98      3561
   macro avg       0.93      0.95      0.94      3561
weighted avg       0.98      0.98      0.98      3561


Confusion Matrix:

[[3190   46]
 [  25  300]]
Recall: 97.25% <-----------------------
[[0.97775031 0.22153846]
 [0.00803461 0.92      ]]
```



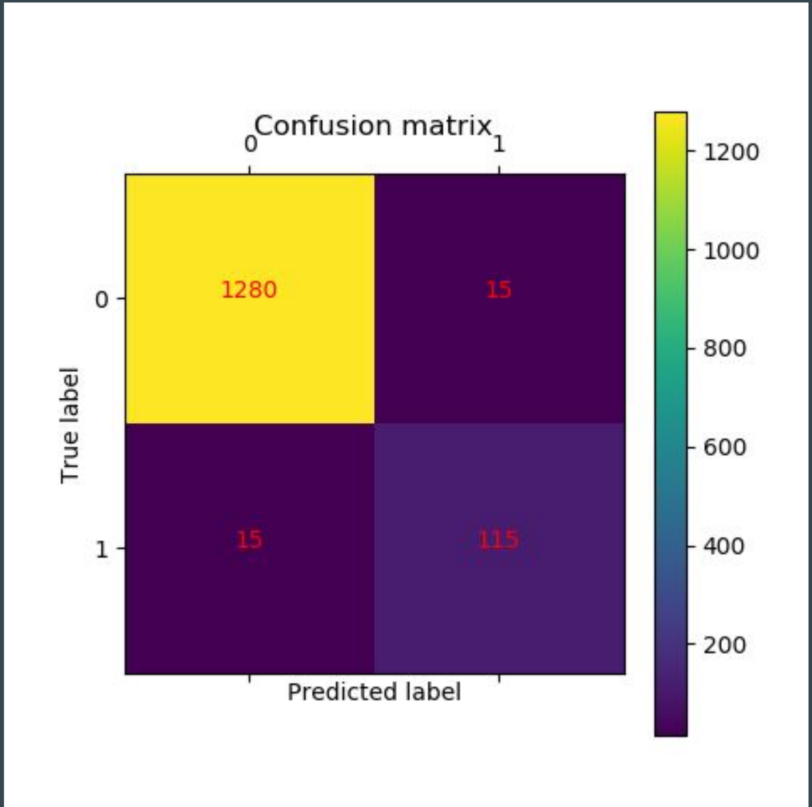
```
Classification Report (val):

              precision    recall  f1-score   support

           0       0.99      0.99      0.99      1295
           1       0.87      0.91      0.89       130

    accuracy                           0.98      1425
   macro avg       0.93      0.95      0.94      1425
weighted avg       0.98      0.98      0.98      1425


Confusion Matrix:

[[1278   17]
 [  12  118]]
Recall: 97.9%
[[0.98841699 0.11538462]
 [0.01158301 0.88461538]]
```

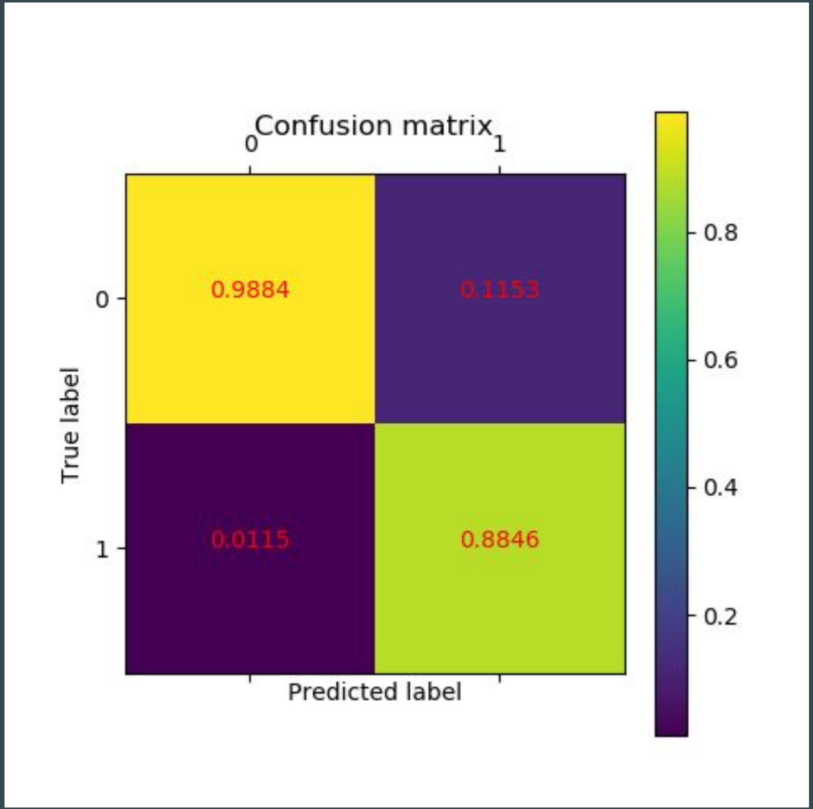
3. Confusion matrix for test set





3. Confusion matrix for validation set



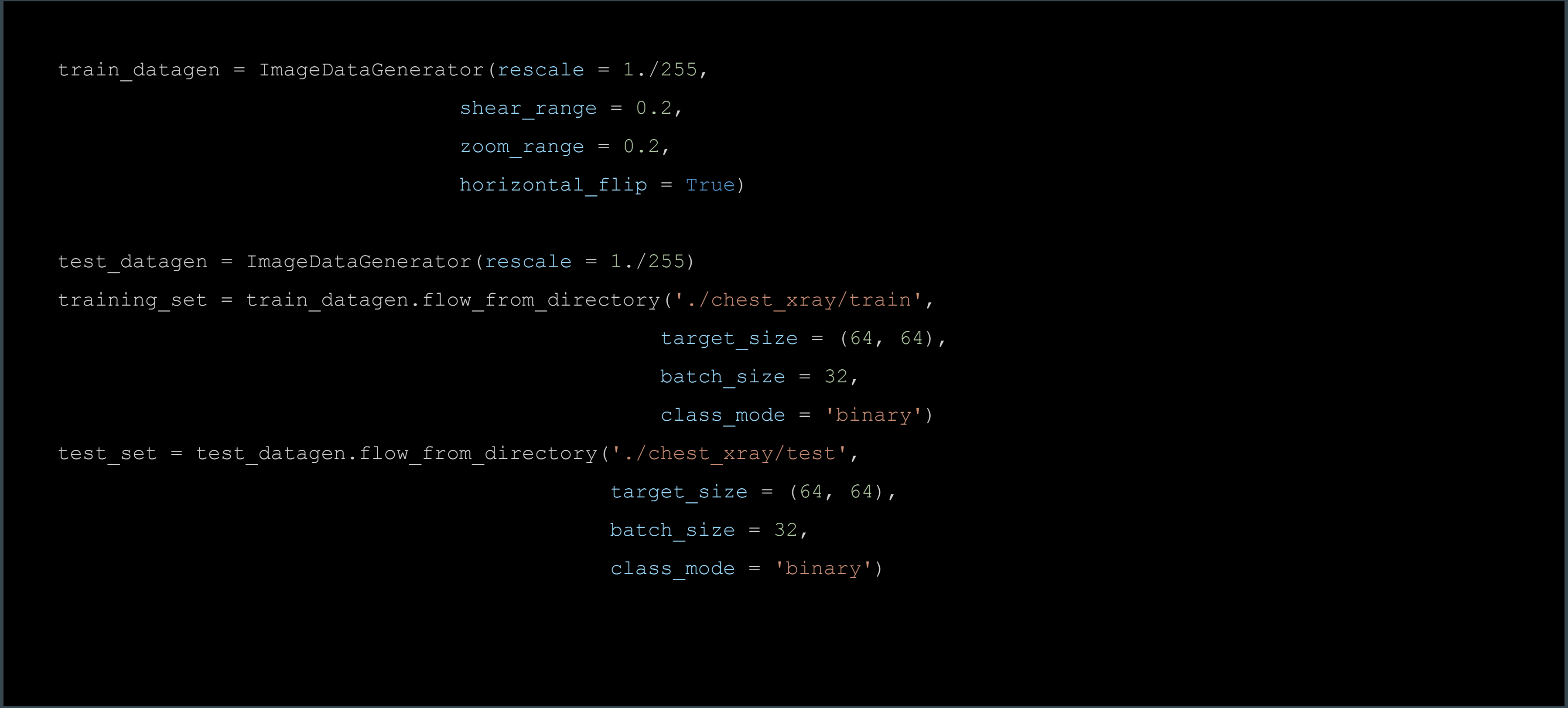


Pneumonia prediction

Using CNN

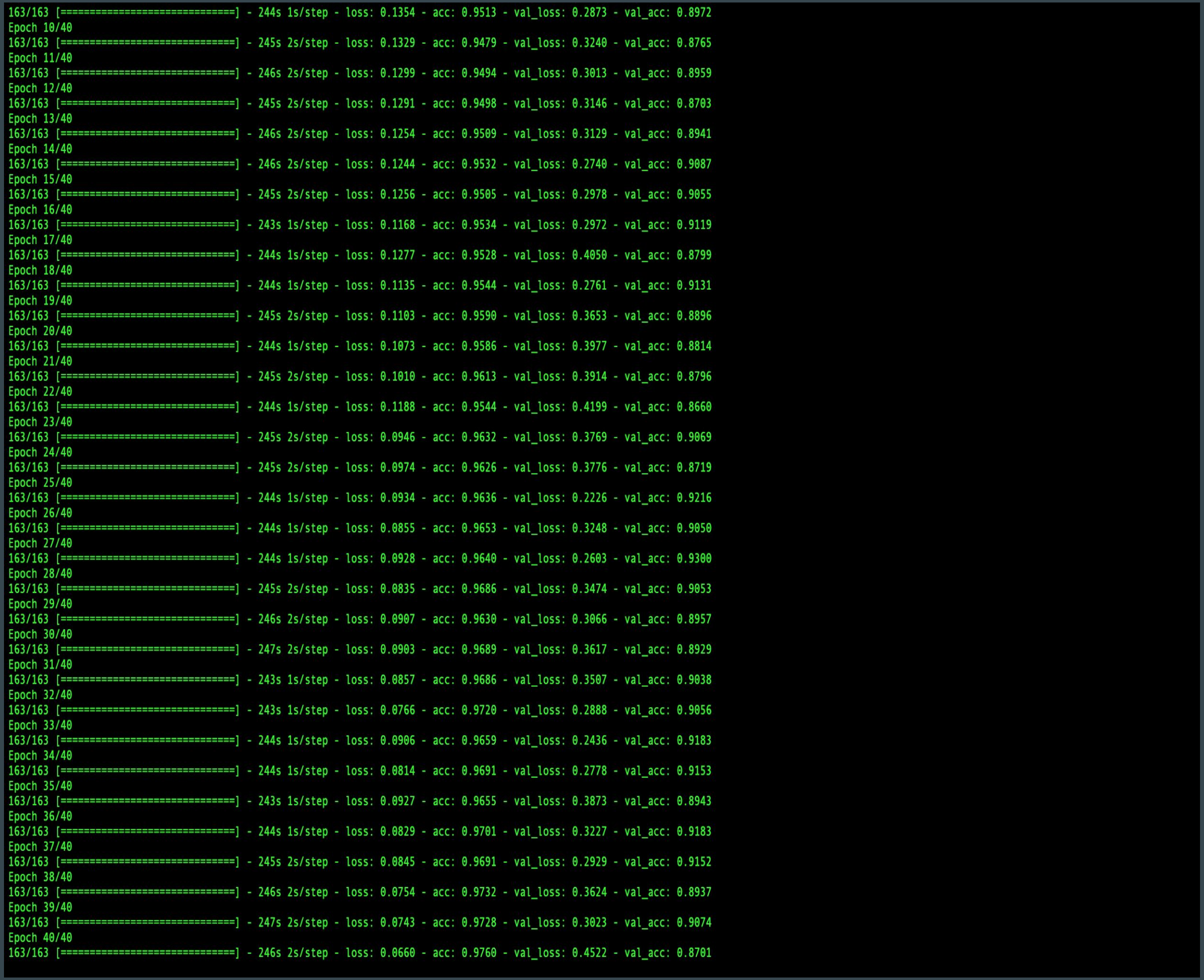
```
# Build the CNN
classifier = Sequential()
# Convolution
classifier.add(Conv2D(32, (3, 3), activation="relu", input_shape=(64, 64, 3)))
# Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Pooling is made with a 2x2 array
# Add 2nd convolutional layer with the same structure as the 1st to improve predictions
classifier.add(Conv2D(32, (3, 3), activation="relu"))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
# Flattening
classifier.add(Flatten())
# Full Connection
classifier.add(Dense(activation = 'relu', units = 128))
classifier.add(Dense(activation = 'sigmoid', units = 1))
# Compile the CNN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

Using Image Data Generator



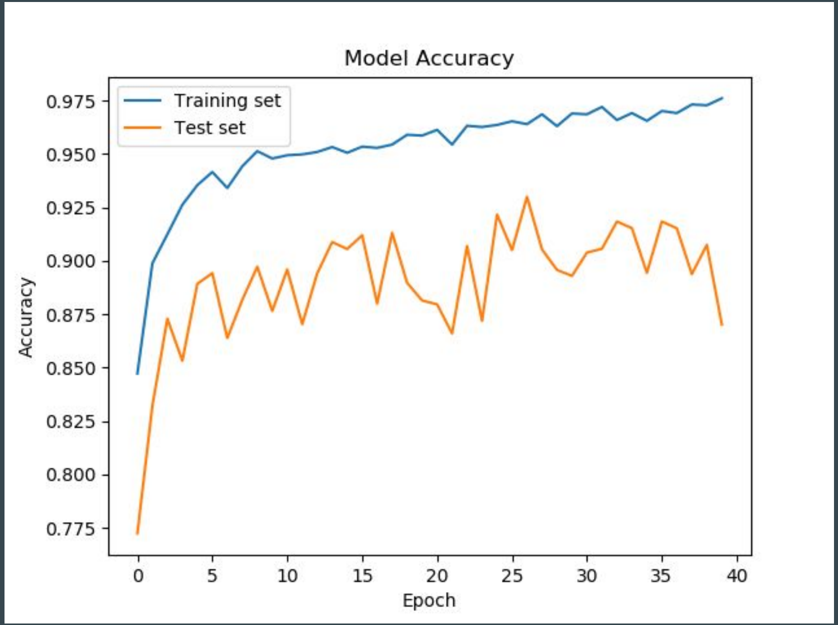
```
train_datagen = ImageDataGenerator(rescale = 1./255,
                                shear_range = 0.2,
                                zoom_range = 0.2,
                                horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
training_set = train_datagen.flow_from_directory('./chest_xray/train',
                                                target_size = (64, 64),
                                                batch_size = 32,
                                                class_mode = 'binary')
test_set = test_datagen.flow_from_directory('./chest_xray/test',
                                            target_size = (64, 64),
                                            batch_size = 32,
                                            class_mode = 'binary')
```

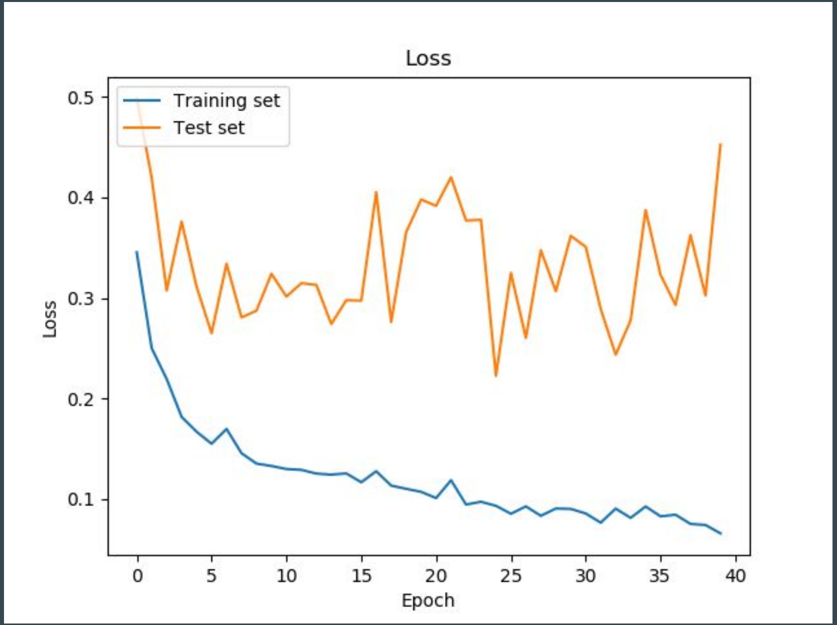
Running on 40 epoch



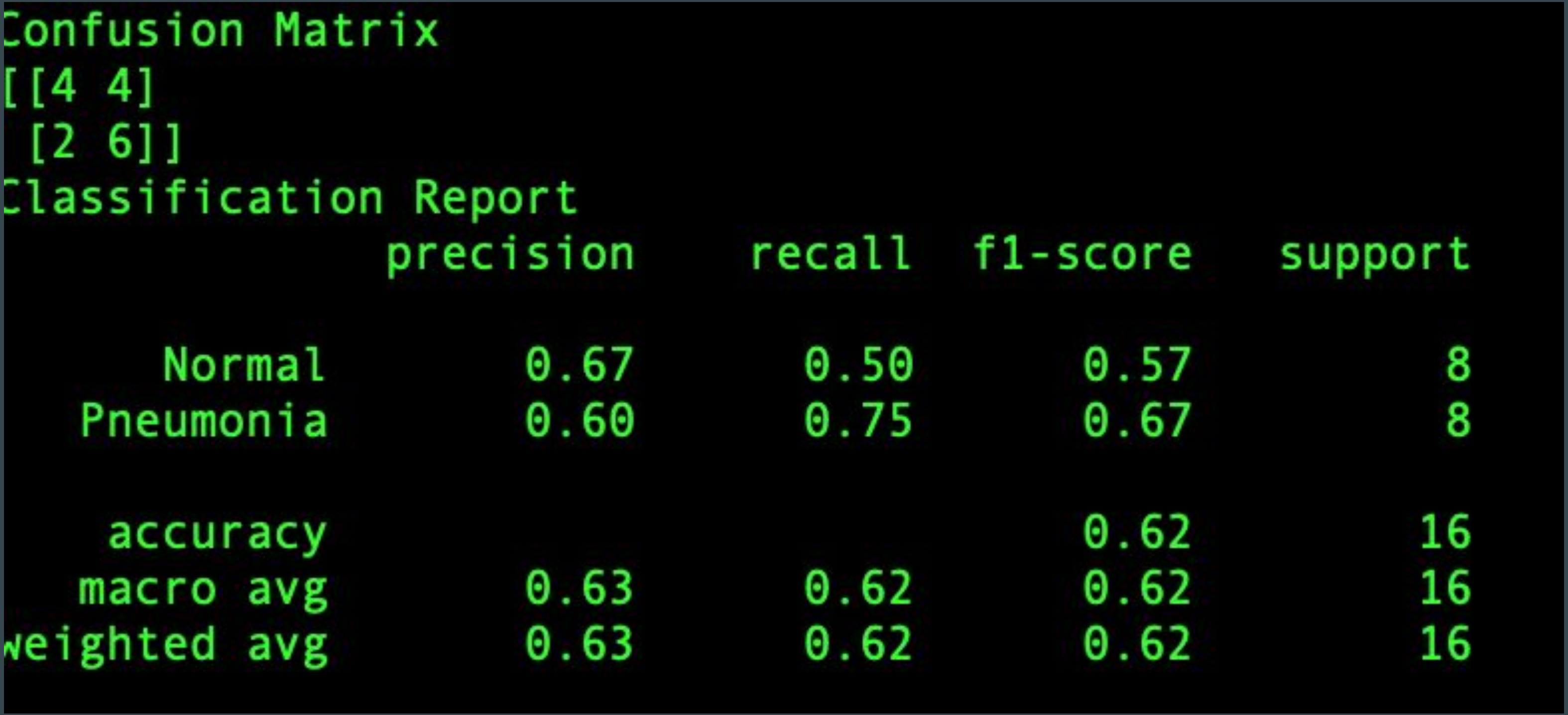
```
163/163 [==============================] - 244s 1s/step - loss: 0.1354 - acc: 0.9513 - val_loss: 0.2873 - val_acc: 0.8972
Epoch 10/40
163/163 [==============================] - 245s 2s/step - loss: 0.1329 - acc: 0.9479 - val_loss: 0.3240 - val_acc: 0.8765
Epoch 11/40
163/163 [==============================] - 246s 2s/step - loss: 0.1299 - acc: 0.9494 - val_loss: 0.3013 - val_acc: 0.8959
Epoch 12/40
163/163 [==============================] - 245s 2s/step - loss: 0.1291 - acc: 0.9498 - val_loss: 0.3146 - val_acc: 0.8703
Epoch 13/40
163/163 [==============================] - 246s 2s/step - loss: 0.1254 - acc: 0.9509 - val_loss: 0.3129 - val_acc: 0.8941
Epoch 14/40
163/163 [==============================] - 246s 2s/step - loss: 0.1244 - acc: 0.9532 - val_loss: 0.2740 - val_acc: 0.9087
Epoch 15/40
163/163 [==============================] - 245s 2s/step - loss: 0.1256 - acc: 0.9505 - val_loss: 0.2978 - val_acc: 0.9055
Epoch 16/40
163/163 [==============================] - 243s 1s/step - loss: 0.1168 - acc: 0.9534 - val_loss: 0.2972 - val_acc: 0.9119
Epoch 17/40
163/163 [==============================] - 244s 1s/step - loss: 0.1277 - acc: 0.9528 - val_loss: 0.4050 - val_acc: 0.8799
Epoch 18/40
163/163 [==============================] - 244s 1s/step - loss: 0.1135 - acc: 0.9544 - val_loss: 0.2761 - val_acc: 0.9131
Epoch 19/40
163/163 [==============================] - 245s 2s/step - loss: 0.1103 - acc: 0.9590 - val_loss: 0.3653 - val_acc: 0.8896
Epoch 20/40
163/163 [==============================] - 244s 1s/step - loss: 0.1073 - acc: 0.9586 - val_loss: 0.3977 - val_acc: 0.8814
Epoch 21/40
163/163 [==============================] - 245s 2s/step - loss: 0.1010 - acc: 0.9613 - val_loss: 0.3914 - val_acc: 0.8796
Epoch 22/40
163/163 [==============================] - 244s 1s/step - loss: 0.1188 - acc: 0.9544 - val_loss: 0.4199 - val_acc: 0.8660
Epoch 23/40
163/163 [==============================] - 245s 2s/step - loss: 0.0946 - acc: 0.9632 - val_loss: 0.3769 - val_acc: 0.9069
Epoch 24/40
163/163 [==============================] - 245s 2s/step - loss: 0.0974 - acc: 0.9626 - val_loss: 0.3776 - val_acc: 0.8719
Epoch 25/40
163/163 [==============================] - 244s 1s/step - loss: 0.0934 - acc: 0.9636 - val_loss: 0.2226 - val_acc: 0.9216
Epoch 26/40
163/163 [==============================] - 244s 1s/step - loss: 0.0855 - acc: 0.9653 - val_loss: 0.3248 - val_acc: 0.9050
Epoch 27/40
163/163 [==============================] - 244s 1s/step - loss: 0.0928 - acc: 0.9640 - val_loss: 0.2603 - val_acc: 0.9300
Epoch 28/40
163/163 [==============================] - 245s 2s/step - loss: 0.0835 - acc: 0.9686 - val_loss: 0.3474 - val_acc: 0.9053
Epoch 29/40
163/163 [==============================] - 246s 2s/step - loss: 0.0907 - acc: 0.9630 - val_loss: 0.3066 - val_acc: 0.8957
Epoch 30/40
163/163 [==============================] - 247s 2s/step - loss: 0.0903 - acc: 0.9689 - val_loss: 0.3617 - val_acc: 0.8929
Epoch 31/40
163/163 [==============================] - 243s 1s/step - loss: 0.0857 - acc: 0.9686 - val_loss: 0.3507 - val_acc: 0.9038
Epoch 32/40
163/163 [==============================] - 243s 1s/step - loss: 0.0766 - acc: 0.9720 - val_loss: 0.2888 - val_acc: 0.9056
Epoch 33/40
163/163 [==============================] - 244s 1s/step - loss: 0.0906 - acc: 0.9659 - val_loss: 0.2436 - val_acc: 0.9183
Epoch 34/40
163/163 [==============================] - 244s 1s/step - loss: 0.0814 - acc: 0.9691 - val_loss: 0.2778 - val_acc: 0.9153
Epoch 35/40
163/163 [==============================] - 243s 1s/step - loss: 0.0927 - acc: 0.9655 - val_loss: 0.3873 - val_acc: 0.8943
Epoch 36/40
163/163 [==============================] - 244s 1s/step - loss: 0.0829 - acc: 0.9701 - val_loss: 0.3227 - val_acc: 0.9183
Epoch 37/40
163/163 [==============================] - 245s 2s/step - loss: 0.0845 - acc: 0.9691 - val_loss: 0.2929 - val_acc: 0.9152
Epoch 38/40
163/163 [==============================] - 246s 2s/step - loss: 0.0754 - acc: 0.9732 - val_loss: 0.3624 - val_acc: 0.8937
Epoch 39/40
163/163 [==============================] - 247s 2s/step - loss: 0.0743 - acc: 0.9728 - val_loss: 0.3023 - val_acc: 0.9074
Epoch 40/40
163/163 [==============================] - 246s 2s/step - loss: 0.0660 - acc: 0.9760 - val_loss: 0.4522 - val_acc: 0.8701
```

On epoch 32 model become stable





Going on model on ep.32 and testing on validation set



```
Confusion Matrix
[[4 4]
 [2 6]]
Classification Report
              precision    recall  f1-score   support

      Normal       0.67      0.50      0.57         8
   Pneumonia       0.60      0.75      0.67         8

    accuracy                           0.62        16
   macro avg       0.63      0.62      0.62        16
weighted avg       0.63      0.62      0.62        16
```