pulsar4

October 8, 2022

1 Pulsar Emission Data Analysis

2 All Imports that may or may not be needed and used for the notebook

[ ]:
```
#currently including any and all Imports that maybe needed for the project.
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.linear_model import LogisticRegression, LinearRegression
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.feature_selection import RFE
import datetime as dt
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances
from scipy.cluster.hierarchy import linkage, dendrogram, cut_tree
from scipy.spatial.distance import pdist
from sklearn.feature_extraction.text import TfidfVectorizer
import matplotlib.dates as mdates
from scipy.stats import pearsonr
from scipy import stats
import statistics
import math
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.tsa.tsatools import lagmat
from numpy import array
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import LSTM
from keras.layers import Dense
from keras.layers import Bidirectional
```

3 Section for extracting from a tar file.

Currently implemented for original TAR File structure.

[ ]:
```
#This is also found in the main file under tarunzip.py
import tarfile
import os
import sys

#tar = tarfile.open("pulseTarFile.tar")
#tar.extractall('./Data')
#tar.close()
```

3.1 Beginning of Exploration

3.1.1 Examining the data

In this section we are determining the total integrity of the data to determine if further comprehensive data cleaning and uniforming processes are needed.

[ ]:
```
colnames = ['Pulse Number', 'Brightness', 'Uncertainty']
pulsar = pd.read_csv("Data/J1243-6423.pulses", sep = ' ', header = None, names␣
↪= colnames)
```

[ ]:
```
pulsar.shape
```

[ ]:
```
(1819, 3)
```

[ ]:
```
pulsar.head(25)
```

[ ]:

	Pulse Number	Brightness	Uncertainty
0	1	0.101127	0.001893
1	2	0.012166	0.001814
2	3	0.021918	0.001835
3	4	0.181179	0.002183
4	5	0.000240	0.001725
5	6	0.085866	0.001723
6	7	0.067280	0.001778
7	8	0.092884	0.002438
8	9	0.083350	0.002101
9	10	0.087871	0.001941
10	11	0.123529	0.002026
11	12	0.097413	0.001878
12	13	0.100649	0.001820
13	14	0.058025	0.001724
14	15	0.116164	0.001948
15	16	0.029203	0.001918
16	17	0.174895	0.002131
17	18	0.200468	0.002571
18	19	0.123890	0.001805

```
19            20   0.083496   0.001856
20            21   0.042757   0.001891
21            22   0.119953   0.001744
22            23   0.096266   0.001911
23            24   0.040698   0.001975
24            25   0.175852   0.002251
```

[ ]:
```
pulsar.describe()
```

[ ]:
```
       Pulse Number   Brightness  Uncertainty
count   1819.000000  1819.000000  1819.000000
mean     910.000000     0.075070     0.001958
std      525.244388     0.057006     0.000306
min        1.000000    -0.004643     0.001532
25%      455.500000     0.019738     0.001774
50%      910.000000     0.076660     0.001872
75%     1364.500000     0.112285     0.002041
max     1819.000000     0.269903     0.005952
```

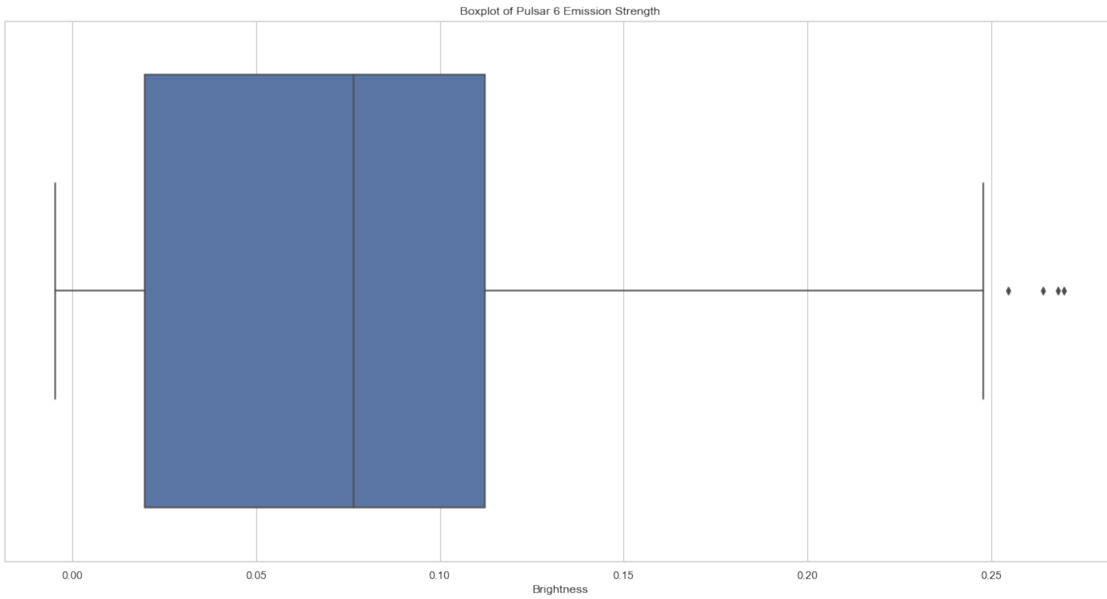
[ ]:
```
nullBoolBrightness = pd.isnull(pulsar["Brightness"])

pulsar[nullBoolBrightness]
```

[ ]:
```
Empty DataFrame
Columns: [Pulse Number, Brightness, Uncertainty]
Index: []
```

[ ]:
```
pulsar["Brightness"].describe()
```

[ ]:
```
count    1819.000000
mean        0.075070
std         0.057006
min        -0.004643
25%         0.019738
50%         0.076660
75%         0.112285
max         0.269903
Name: Brightness, dtype: float64
```

[ ]:
```
plt.figure(figsize=(20,10))
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=pulsar["Brightness"]).set_title("Boxplot of Pulsar 6␣
↪Emission Strength")
```



[ ]:
```
medianpulse6 = pulsar["Brightness"].median()
print("Median of Pulsar6: ", medianpulse6)
pulsar['Binary'] = np.where(pulsar['Brightness'] > medianpulse6, 1, 0)
```

```
Median of Pulsar6:  0.07665979
```

[ ]:
```
pulsar
```

[ ]:

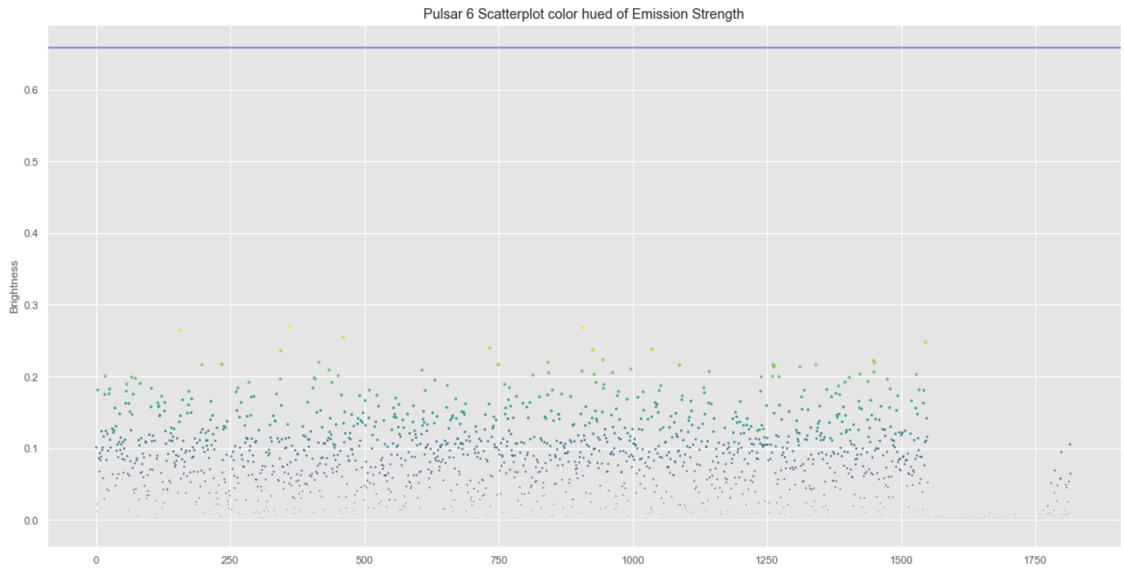
	Pulse Number	Brightness	Uncertainty	Binary
0	1	0.101127	0.001893	1
1	2	0.012166	0.001814	0
2	3	0.021918	0.001835	0
3	4	0.181179	0.002183	1
4	5	0.000240	0.001725	0
…	…	…	…	…
1814	1815	0.105178	0.002086	1
1815	1816	0.064272	0.001995	0
1816	1817	0.000171	0.001730	0
1817	1818	-0.000924	0.001706	0
1818	1819	0.000001	0.001532	0

[1819 rows x 4 columns]

[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = pulsar.Brightness.values
plt.style.use('ggplot')
```

```
ax = sns.scatterplot(data=pulsar["Brightness"], s= strength*50, c=strength,␣
↪cmap="viridis", marker="o").set_title('Pulsar 6 Scatterplot color hued of␣
↪Emission Strength')
ax= plt.axhline( y=0.07665979, ls='-',c='mediumslateblue')
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\matplotlib\collections.py:1003:
RuntimeWarning: invalid value encountered in sqrt
  scale = np.sqrt(self._sizes) * dpi / 72.0 * self._factor
```

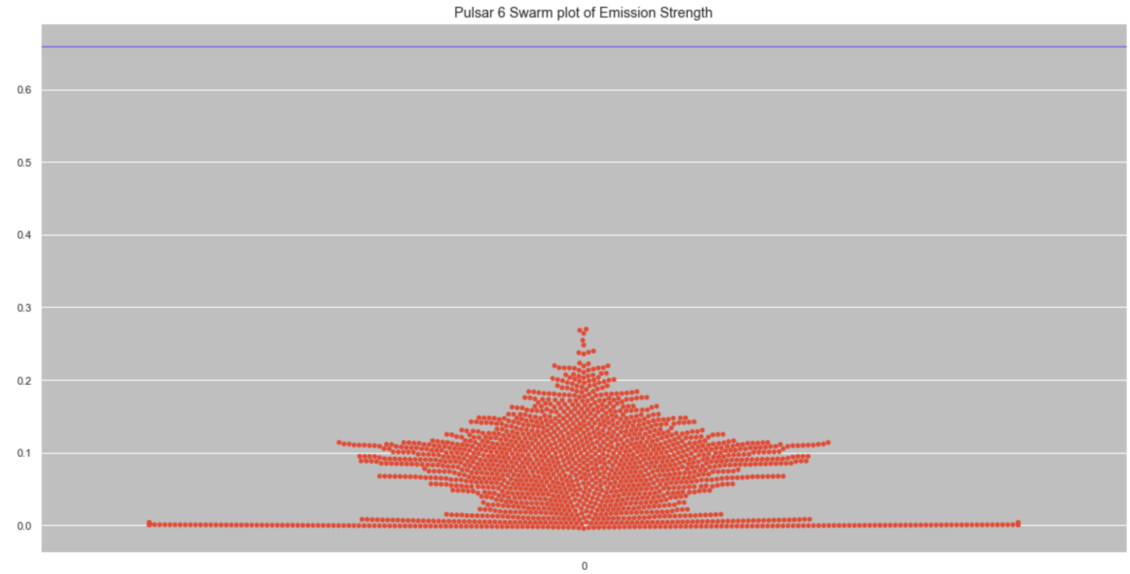



```
[ ]: print(len(pulsar[(pulsar.Brightness > 0.07665979)]))
     print(len(pulsar[(pulsar.Brightness < 0.07665979)]))
```

```
0
1819
```

```
[ ]: plt.figure(figsize=(20,10))
     sns.set_style("darkgrid", {"axes.facecolor": ".75"})
     strength = pulsar.Brightness.values
     ax = plt.axhline( y=0.07665979, ls='-',c='mediumslateblue')
     ax = sns.swarmplot(data=pulsar["Brightness"], c="blue").set_title('Pulsar 6␣
     ↪Swarm plot of Emission Strength')
```

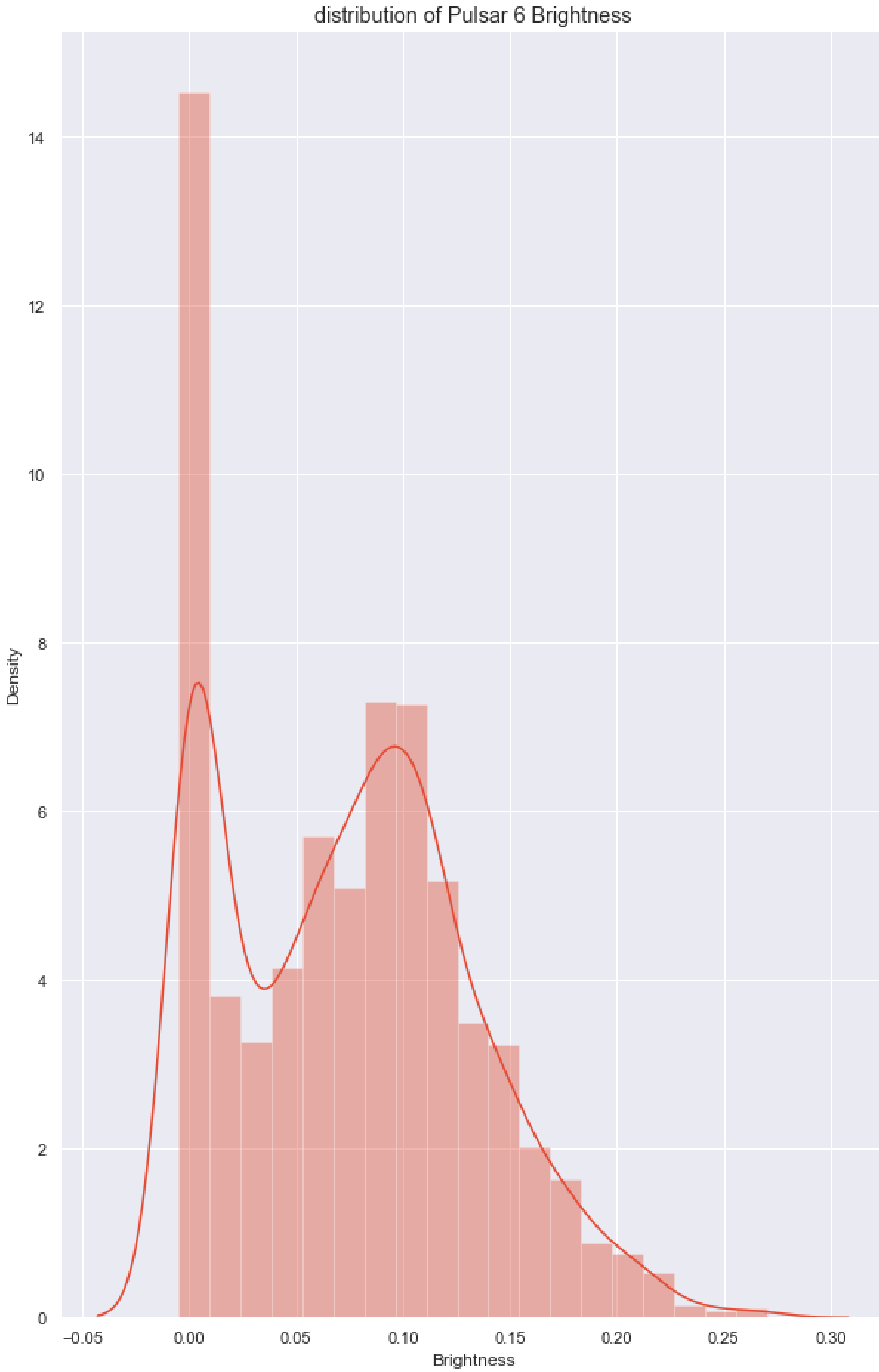
```
c:\Users\oxlay\anaconda3\lib\site-packages\seaborn\categorical.py:1296:
UserWarning: 5.7% of the points cannot be placed; you may want to decrease the
size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



[ ]:
```
plt.figure(figsize=(10, 16))
with sns.axes_style('darkgrid'):
    sns.distplot(pulsar.Brightness)
plt.title("distribution of Pulsar 6 Brightness")
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

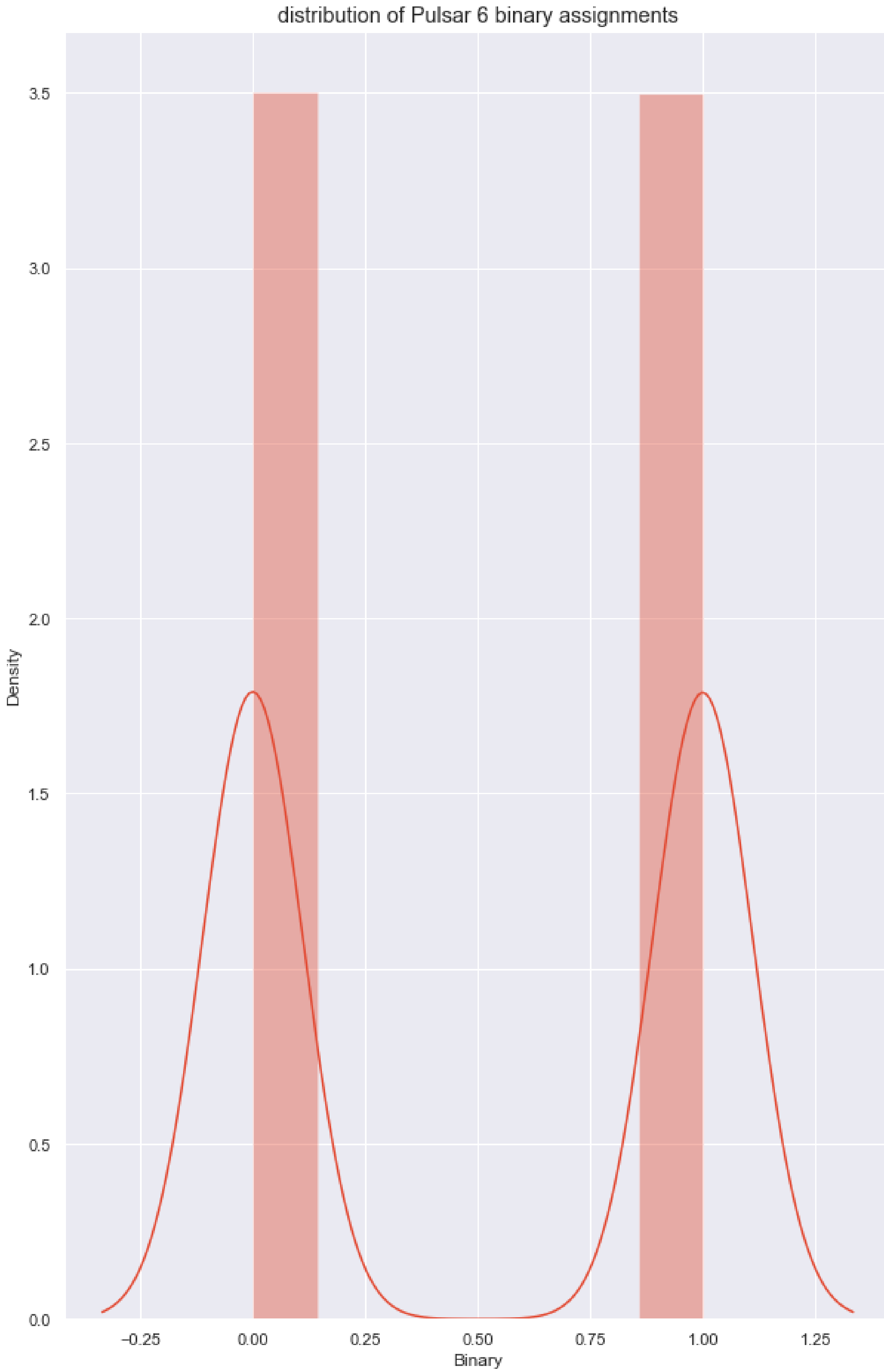
[ ]:
```
Text(0.5, 1.0, 'distribution of Pulsar 6 Brightness')
```


distribution of Pulsar 6 Brightness
Density
Brightness
0
2
4
6
8
10
12
14
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30

[ ]:
```python
plt.figure(figsize=(10, 16))
with sns.axes_style('darkgrid'):
    sns.distplot(pulsar.Binary)
plt.title("distribution of Pulsar 6 binary assignments")
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

[ ]:
```
Text(0.5, 1.0, 'distribution of Pulsar 6 binary assignments')
```


distribution of Pulsar 6 binary assignments
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
Binary

3.2 Binary Classification

[ ]:
```python
X = pulsar[['Brightness', 'Uncertainty']]
y = pulsar['Binary']
```

[ ]:
```python
X.head()
```

[ ]:

	Brightness	Uncertainty
0	0.101127	0.001893
1	0.012166	0.001814
2	0.021918	0.001835
3	0.181179	0.002183
4	0.000240	0.001725

[ ]:
```python
y.head()
```

[ ]:
```
0    1
1    0
2    0
3    1
4    0
Name: Binary, dtype: int32
```

[ ]:
```python
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y , test_size=0.20)
```

[ ]:
```python
from sklearn.preprocessing import StandardScaler

train_scaler = StandardScaler()
X_train = train_scaler.fit_transform(X_train)

test_scaler = StandardScaler()
X_test = test_scaler.fit_transform(X_test)
```

[ ]:
```python
model = LogisticRegression()

model.fit(X_train, y_train)
```

[ ]:
```
LogisticRegression()
```

[ ]:
```python
predictions = model.predict(X_test)
```

[ ]:
```python
from sklearn.metrics import confusion_matrix
```

```
cm = confusion_matrix(y_test, predictions)

TN, FP, FN, TP = confusion_matrix(y_test, predictions).ravel()

print('True Positive(TP)  = ', TP)
print('False Positive(FP) = ', FP)
print('True Negative(TN)  = ', TN)
print('False Negative(FN) = ', FN)
```

```
True Positive(TP)  =  166
False Positive(FP) =  15
True Negative(TN)  =  183
False Negative(FN) =  0
```

[ ]:
```
accuracy =  (TP + TN) / (TP + FP + TN + FN)

print("Accuracy of the model is ", accuracy)
```

```
Accuracy of the model is  0.9587912087912088
```

3.3 Bidirectional LSTM Model

[ ]:
```
brightness_list = list(pulsar['Brightness'])
brightness_list[:10]
```

[ ]:
```
[0.1011271,
 0.01216605,
 0.02191846,
 0.1811794,
 0.0002404589,
 0.08586562,
 0.06727986,
 0.09288353,
 0.08335005,
 0.08787134]
```

[ ]:
```
def split_list(blist, steps):
    X, y = list(), list()
    for i in range(len(blist)):
        # find the end of this pattern
        end_ix = i + steps
        # check if we are beyond the sequence
        if end_ix > len(blist)-1:
            break
        # gather input and output parts of the pattern
        list_x, list_y = blist[i:end_ix], blist[end_ix]
        X.append(list_x)
        y.append(list_y)
```

```
    return array(X), array(y)
```

[ ]:
```
X, y = split_list(brightness_list, 100)
X = X.reshape((X.shape[0], X.shape[1], 1))
X[:1]
```

[ ]:
```
array([[[ 0.1011271 ],
        [ 0.01216605],
        [ 0.02191846],
        [ 0.1811794 ],
        [ 0.00024046],
        [ 0.08586562],
        [ 0.06727986],
        [ 0.09288353],
        [ 0.08335005],
        [ 0.08787134],
        [ 0.1235293 ],
        [ 0.09741315],
        [ 0.1006486 ],
        [ 0.05802517],
        [ 0.1161637 ],
        [ 0.02920314],
        [ 0.1748954 ],
        [ 0.2004684 ],
        [ 0.1238901 ],
        [ 0.08349596],
        [ 0.04275741],
        [ 0.119953  ],
        [ 0.09626626],
        [ 0.04069848],
        [ 0.1758523 ],
        [ 0.1823138 ],
        [ 0.111629  ],
        [ 0.100765  ],
        [ 0.0011859 ],
        [ 0.00056381],
        [ 0.1270942 ],
        [ 0.0635269 ],
        [ 0.1217607 ],
        [ 0.1248414 ],
        [-0.00045107],
        [ 0.1558657 ],
        [ 0.04782556],
        [ 0.1304564 ],
        [ 0.09637598],
        [ 0.05783188],
        [ 0.1050685 ],
```

```
[ 0.08733439],
[ 0.06381079],
[ 0.1083339 ],
[ 0.1478652 ],
[ 0.09749309],
[ 0.1146415 ],
[ 0.1052436 ],
[ 0.09187137],
[ 0.08488005],
[ 0.1085781 ],
[ 0.06562565],
[ 0.09941817],
[ 0.0838001 ],
[ 0.1117044 ],
[ 0.1626449 ],
[ 0.1793863 ],
[ 0.1892244 ],
[ 0.08889077],
[ 0.06400782],
[ 0.01289888],
[ 0.1623591 ],
[ 0.06149212],
[ 0.1508985 ],
[ 0.08169965],
[ 0.1249356 ],
[ 0.1989626 ],
[ 0.1469875 ],
[ 0.07229389],
[ 0.1752627 ],
[ 0.04166696],
[ 0.03740181],
[ 0.07867823],
[ 0.1974833 ],
[ 0.04297894],
[ 0.0828494 ],
[ 0.08564399],
[ 0.00509527],
[ 0.00545146],
[ 0.1110586 ],
[ 0.02850289],
[ 0.1037645 ],
[ 0.1901705 ],
[ 0.06521289],
[ 0.1137405 ],
[ 0.06645427],
[ 0.07760051],
[ 0.09210339],
```

```
        [ 0.05927911],
        [ 0.08705323],
        [ 0.04752931],
        [-0.00038894],
        [ 0.1090507 ],
        [ 0.1077409 ],
        [ 0.02959244],
        [ 0.03416125],
        [ 0.06223557],
        [ 0.04000795],
        [ 0.03429767],
        [ 0.1185424 ]]])
```

[ ]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y , test_size=0.20)
```

[ ]:
```
model = Sequential()
model.add(Bidirectional(LSTM(50, activation='relu'), input_shape=(100, 1)))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',␣
 ↪metrics=['accuracy'])
```

[ ]:
```
history = model.fit(X_train, y_train, epochs=50, verbose=1, batch_size=10)
```

```
Epoch 1/200
138/138 [==============================] - 4s 21ms/step - loss: 0.7254 -
accuracy: 0.0000e+00
Epoch 2/200
138/138 [==============================] - 3s 21ms/step - loss: 0.5564 -
accuracy: 0.0000e+00
Epoch 3/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2953 -
accuracy: 0.0000e+00
Epoch 4/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2704 -
accuracy: 0.0000e+00
Epoch 5/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2684 -
accuracy: 0.0000e+00
Epoch 6/200
138/138 [==============================] - 3s 24ms/step - loss: 0.2650 -
accuracy: 0.0000e+00
Epoch 7/200
138/138 [==============================] - 3s 23ms/step - loss: 0.2578 -
accuracy: 0.0000e+00
Epoch 8/200
138/138 [==============================] - 3s 23ms/step - loss: 0.2590 -
accuracy: 0.0000e+00
```

```
Epoch 9/200
138/138 [==============================] - 4s 26ms/step - loss: 0.2576 -
accuracy: 0.0000e+00
Epoch 10/200
138/138 [==============================] - 4s 27ms/step - loss: 0.2569 -
accuracy: 0.0000e+00
Epoch 11/200
138/138 [==============================] - 3s 24ms/step - loss: 0.2566 -
accuracy: 0.0000e+00
Epoch 12/200
138/138 [==============================] - 4s 30ms/step - loss: 0.2572 -
accuracy: 0.0000e+00
Epoch 13/200
138/138 [==============================] - 4s 27ms/step - loss: 0.2564 -
accuracy: 0.0000e+00
Epoch 14/200
138/138 [==============================] - 3s 25ms/step - loss: 0.2573 -
accuracy: 0.0000e+00
Epoch 15/200
138/138 [==============================] - 3s 25ms/step - loss: 0.2560 -
accuracy: 0.0000e+00
Epoch 16/200
138/138 [==============================] - 4s 29ms/step - loss: 0.2648 -
accuracy: 0.0000e+00
Epoch 17/200
138/138 [==============================] - 3s 24ms/step - loss: 0.2648 -
accuracy: 0.0000e+00
Epoch 18/200
138/138 [==============================] - 3s 23ms/step - loss: 0.2796 -
accuracy: 0.0000e+00
Epoch 19/200
138/138 [==============================] - 4s 26ms/step - loss: 0.2747 -
accuracy: 0.0000e+00
Epoch 20/200
138/138 [==============================] - 4s 26ms/step - loss: 0.2653 -
accuracy: 0.0000e+00
Epoch 21/200
138/138 [==============================] - 3s 23ms/step - loss: 0.2651 -
accuracy: 0.0000e+00
Epoch 22/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2651 -
accuracy: 0.0000e+00
Epoch 23/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2650 -
accuracy: 0.0000e+00
Epoch 24/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2649 -
accuracy: 0.0000e+00
```

```
Epoch 25/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2648 -
accuracy: 0.0000e+00
Epoch 26/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2647 -
accuracy: 0.0000e+00
Epoch 27/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2644 -
accuracy: 0.0000e+00
Epoch 28/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2776 -
accuracy: 0.0000e+00
Epoch 29/200
138/138 [==============================] - 3s 21ms/step - loss: 1437904896.0000
- accuracy: 0.0000e+00
Epoch 30/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2649 -
accuracy: 0.0000e+00
Epoch 31/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2646 -
accuracy: 0.0000e+00
Epoch 32/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2645 -
accuracy: 0.0000e+00
Epoch 33/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2644 -
accuracy: 0.0000e+00
Epoch 34/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2643 -
accuracy: 0.0000e+00
Epoch 35/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2643 -
accuracy: 0.0000e+00
Epoch 36/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2642 -
accuracy: 0.0000e+00
Epoch 37/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2641 -
accuracy: 0.0000e+00
Epoch 38/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2640 -
accuracy: 0.0000e+00
Epoch 39/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2638 -
accuracy: 0.0000e+00
Epoch 40/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2634 -
accuracy: 0.0000e+00
```

```
Epoch 41/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2622 -
accuracy: 0.0000e+00
Epoch 42/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2618 -
accuracy: 0.0000e+00
Epoch 43/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2619 -
accuracy: 0.0000e+00
Epoch 44/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2612 -
accuracy: 0.0000e+00
Epoch 45/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2608 -
accuracy: 0.0000e+00
Epoch 46/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2605 -
accuracy: 0.0000e+00
Epoch 47/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2603 -
accuracy: 0.0000e+00
Epoch 48/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2603 -
accuracy: 0.0000e+00
Epoch 49/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2599 -
accuracy: 0.0000e+00
Epoch 50/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2596 -
accuracy: 0.0000e+00
Epoch 51/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2592 -
accuracy: 0.0000e+00
Epoch 52/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2594 -
accuracy: 0.0000e+00
Epoch 53/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2586 -
accuracy: 0.0000e+00
Epoch 54/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2586 -
accuracy: 0.0000e+00
Epoch 55/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2585 -
accuracy: 0.0000e+00
Epoch 56/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2583 -
accuracy: 0.0000e+00
```

```
Epoch 57/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2580 -
accuracy: 0.0000e+00
Epoch 58/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2579 -
accuracy: 0.0000e+00
Epoch 59/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2575 -
accuracy: 0.0000e+00
Epoch 60/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2576 -
accuracy: 0.0000e+00
Epoch 61/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2574 -
accuracy: 0.0000e+00
Epoch 62/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2576 -
accuracy: 0.0000e+00
Epoch 63/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2570 -
accuracy: 0.0000e+00
Epoch 64/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2572 -
accuracy: 0.0000e+00
Epoch 65/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2571 -
accuracy: 0.0000e+00
Epoch 66/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2574 -
accuracy: 0.0000e+00
Epoch 67/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2574 -
accuracy: 0.0000e+00
Epoch 68/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2573 -
accuracy: 0.0000e+00
Epoch 69/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2568 -
accuracy: 0.0000e+00
Epoch 70/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2571 -
accuracy: 0.0000e+00
Epoch 71/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2570 -
accuracy: 0.0000e+00
Epoch 72/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2571 -
accuracy: 0.0000e+00
```

```
Epoch 73/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2570 -
accuracy: 0.0000e+00
Epoch 74/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2569 -
accuracy: 0.0000e+00
Epoch 75/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2567 -
accuracy: 0.0000e+00
Epoch 76/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2564 -
accuracy: 0.0000e+00
Epoch 77/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2565 -
accuracy: 0.0000e+00
Epoch 78/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2566 -
accuracy: 0.0000e+00
Epoch 79/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2563 -
accuracy: 0.0000e+00
Epoch 80/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2566 -
accuracy: 0.0000e+00
Epoch 81/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2563 -
accuracy: 0.0000e+00
Epoch 82/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2564 -
accuracy: 0.0000e+00
Epoch 83/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2564 -
accuracy: 0.0000e+00
Epoch 84/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2562 -
accuracy: 0.0000e+00
Epoch 85/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2564 -
accuracy: 0.0000e+00
Epoch 86/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2563 -
accuracy: 0.0000e+00
Epoch 87/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2562 -
accuracy: 0.0000e+00
Epoch 88/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2561 -
accuracy: 0.0000e+00
```

```
Epoch 89/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2562 -
accuracy: 0.0000e+00
Epoch 90/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2563 -
accuracy: 0.0000e+00
Epoch 91/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2562 -
accuracy: 0.0000e+00
Epoch 92/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2560 -
accuracy: 0.0000e+00
Epoch 93/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2561 -
accuracy: 0.0000e+00
Epoch 94/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2561 -
accuracy: 0.0000e+00
Epoch 95/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2560 -
accuracy: 0.0000e+00
Epoch 96/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2558 -
accuracy: 0.0000e+00
Epoch 97/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2560 -
accuracy: 0.0000e+00
Epoch 98/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2560 -
accuracy: 0.0000e+00
Epoch 99/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2560 -
accuracy: 0.0000e+00
Epoch 100/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2558 -
accuracy: 0.0000e+00
Epoch 101/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2559 -
accuracy: 0.0000e+00
Epoch 102/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2559 -
accuracy: 0.0000e+00
Epoch 103/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2557 -
accuracy: 0.0000e+00
Epoch 104/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2554 -
accuracy: 0.0000e+00
```

```
Epoch 105/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2557 -
accuracy: 0.0000e+00
Epoch 106/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2558 -
accuracy: 0.0000e+00
Epoch 107/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2559 -
accuracy: 0.0000e+00
Epoch 108/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2557 -
accuracy: 0.0000e+00
Epoch 109/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2558 -
accuracy: 0.0000e+00
Epoch 110/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2556 -
accuracy: 0.0000e+00
Epoch 111/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2553 -
accuracy: 0.0000e+00
Epoch 112/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2554 -
accuracy: 0.0000e+00
Epoch 113/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2556 -
accuracy: 0.0000e+00
Epoch 114/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2558 -
accuracy: 0.0000e+00
Epoch 115/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2556 -
accuracy: 0.0000e+00
Epoch 116/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2554 -
accuracy: 0.0000e+00
Epoch 117/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2556 -
accuracy: 0.0000e+00
Epoch 118/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2553 -
accuracy: 0.0000e+00
Epoch 119/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2558 -
accuracy: 0.0000e+00
Epoch 120/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2555 -
accuracy: 0.0000e+00
```

```
Epoch 121/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2556 -
accuracy: 0.0000e+00
Epoch 122/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2556 -
accuracy: 0.0000e+00
Epoch 123/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2554 -
accuracy: 0.0000e+00
Epoch 124/200
138/138 [==============================] - 3s 22ms/step - loss: 0.2555 -
accuracy: 0.0000e+00
Epoch 125/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2555 -
accuracy: 0.0000e+00
Epoch 126/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2550 -
accuracy: 0.0000e+00
Epoch 127/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2555 -
accuracy: 0.0000e+00
Epoch 128/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2552 -
accuracy: 0.0000e+00
Epoch 129/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2550 -
accuracy: 0.0000e+00
Epoch 130/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2550 -
accuracy: 0.0000e+00
Epoch 131/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2553 -
accuracy: 0.0000e+00
Epoch 132/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2550 -
accuracy: 0.0000e+00
Epoch 133/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2550 -
accuracy: 0.0000e+00
Epoch 134/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2551 -
accuracy: 0.0000e+00
Epoch 135/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2551 -
accuracy: 0.0000e+00
Epoch 136/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2551 -
accuracy: 0.0000e+00
```

```
Epoch 137/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2549 -
accuracy: 0.0000e+00
Epoch 138/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 139/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2551 -
accuracy: 0.0000e+00
Epoch 140/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 141/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2550 -
accuracy: 0.0000e+00
Epoch 142/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2551 -
accuracy: 0.0000e+00
Epoch 143/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 144/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 145/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 146/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 147/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 148/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 149/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2550 -
accuracy: 0.0000e+00
Epoch 150/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 151/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2549 -
accuracy: 0.0000e+00
Epoch 152/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
```

```
Epoch 153/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 154/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 155/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 156/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 157/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 158/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 159/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 160/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 161/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2546 -
accuracy: 0.0000e+00
Epoch 162/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2546 -
accuracy: 0.0000e+00
Epoch 163/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 164/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2547 -
accuracy: 0.0000e+00
Epoch 165/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2543 -
accuracy: 0.0000e+00
Epoch 166/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 167/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2546 -
accuracy: 0.0000e+00
Epoch 168/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2546 -
accuracy: 0.0000e+00
```

```
Epoch 169/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 170/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 171/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 172/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 173/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 174/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 175/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 176/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2546 -
accuracy: 0.0000e+00
Epoch 177/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2546 -
accuracy: 0.0000e+00
Epoch 178/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 179/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2543 -
accuracy: 0.0000e+00
Epoch 180/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 181/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2548 -
accuracy: 0.0000e+00
Epoch 182/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2542 -
accuracy: 0.0000e+00
Epoch 183/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 184/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
```

```
Epoch 185/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 186/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 187/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2543 -
accuracy: 0.0000e+00
Epoch 188/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2543 -
accuracy: 0.0000e+00
Epoch 189/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2543 -
accuracy: 0.0000e+00
Epoch 190/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 191/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2542 -
accuracy: 0.0000e+00
Epoch 192/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2546 -
accuracy: 0.0000e+00
Epoch 193/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2542 -
accuracy: 0.0000e+00
Epoch 194/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2543 -
accuracy: 0.0000e+00
Epoch 195/200
138/138 [==============================] - 3s 21ms/step - loss: 0.2543 -
accuracy: 0.0000e+00
Epoch 196/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2541 -
accuracy: 0.0000e+00
Epoch 197/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2541 -
accuracy: 0.0000e+00
Epoch 198/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2545 -
accuracy: 0.0000e+00
Epoch 199/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2544 -
accuracy: 0.0000e+00
Epoch 200/200
138/138 [==============================] - 3s 20ms/step - loss: 0.2542 -
accuracy: 0.0000e+00
```

```
[ ]: y_pred = model.predict(X_test, verbose=0)
     y_pred[:10]
```

```
[ ]: array([[0.07059324],
            [0.08830673],
            [0.06158476],
            [0.06819976],
            [0.00528174],
            [0.07549963],
            [0.06386006],
            [0.07609943],
            [0.08484919],
            [0.0804552 ]], dtype=float32)
```

```
[ ]: model.evaluate(X_test, y_test)
```

```
11/11 [==============================] - 0s 7ms/step - loss: 0.2530 - accuracy:
0.0000e+00
```

```
[ ]: [0.25299397110939026, 0.0]
```

3.4 ML Evaluation.

3.4.1 Logistic Regression

This model appears to have gained some insight in the data and accurately defined a majority of the data. The accuracy of the model is >95% which indicates that it was able to determine a trend and apply it in a useful manner in the predictions during evaluation. Further, the confusion matrix further supports the high accuracy and likely usefulness of the model with only 3 false assignments. However, in analysis this is only to determine if there is a correlation between binary assignment and the emission strength x error in measurement. This doesn't aid us in our overall randomness determination, but it does determine that uncertainty has a role in the binary assignment and the overall trust of emission strength.

3.4.2 Bidirectional LSTM

This model is very error prone as the loss value is consistently at 60& or higher at every epoch during training and at exactly 63.07% in evaluation with a 0% accuracy this indicates that there is either a great error in the formation of the model, data used or trend being obtained. Alternatively it could indicate that there is no trend there to predict. Likely this indicates that the model is not valuable for any meaningful analysis.

4 Preliminary runs test

4.0.1 Math Logic

$$Z = \frac{R - \tilde{R}}{s_R}$$

$$\tilde{R} = \frac{2_{n1n2}}{n1 + n2} + 1$$

$$s_R^2 = \frac{2_{n1n2}(2n1n2 - n1 - n2)}{(n1 + n2)^2(n1 + n2 - 1)}$$

link to resource: https://www.geeksforgeeks.org/runs-test-of-randomness-in-python/

$ Z_{critical} = 1.96 $ as the confidence interval level of 95% thus this is a 2 tailed test. If the probability as corrosponding to this confidence interval $ H_{null} $ will be rejected as it is not statistically significant as denoted by $|Z| > Z_{critical} $

There is also code attempting to change it from a z-score probability to a P-score for ease of understanding and clarity.

5 FUNCTION CODE FOR RUNS TEST

[ ]:
```
binaryData1 = pulsar['Binary'].tolist()
print("pulsar6 original: ",binaryData1)
```

```
pulsar6 original:  [1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1,
0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1,
1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1,
0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1,
0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1,
1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0,
1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1,
1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1,
0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,
1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1,
1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0,
1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1,
0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1,
0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0,
0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1,
1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1,
1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0,
0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,
1, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0,
0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0,
1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0,
```

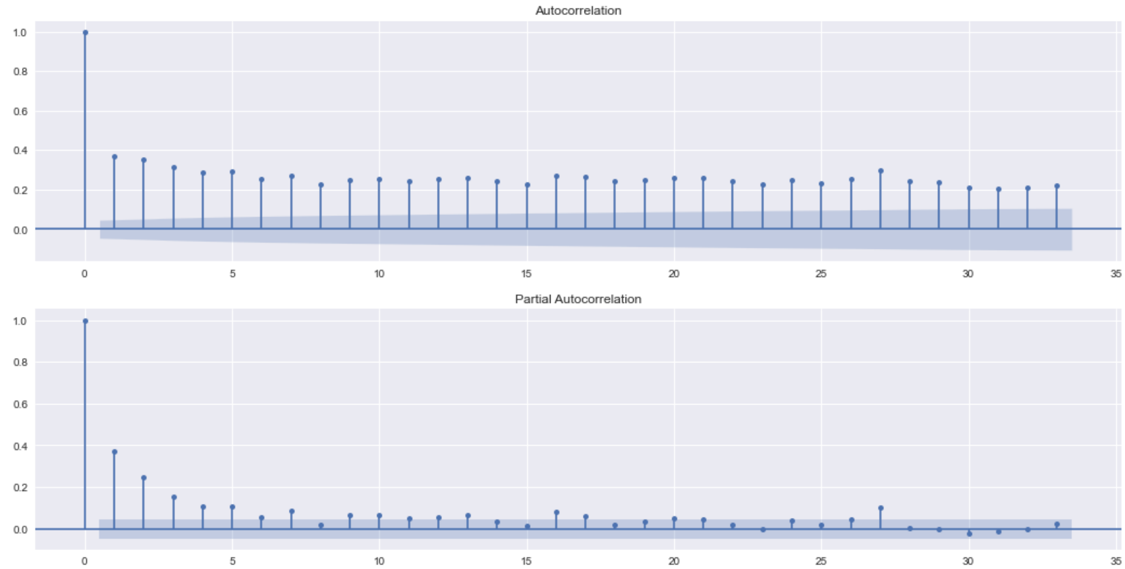
```
1, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1,
1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0,
1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1,
0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1,
1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,
1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0,
0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1,
1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1,
1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0,
1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0,
0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0,
0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0,
0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1,
0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1,
1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1,
0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1,
0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1,
1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1,
1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1,
1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0,
1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1,
0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1,
1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
```

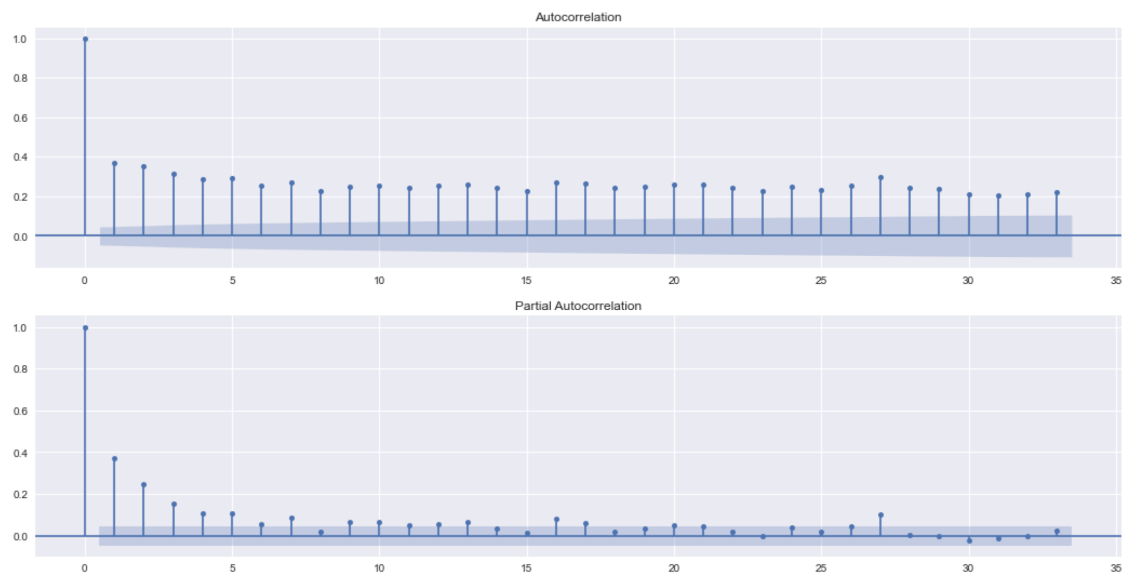
6 Below we begin autocorrelation and autocovariance analysis

To get started with this I am playing around with guide from: https://towardsdatascience.com/a-step-by-step-guide-to-calculating-autocorrelation-and-partial-autocorrelation-8c4342b784e8

[ ]:
```python
plt.style.use("seaborn")
plt.rcParams["figure.figsize"] = (18, 9)

fig, ax = plt.subplots(2,1)

plot_acf(pulsar['Brightness'], ax=ax[0])
plot_pacf(pulsar['Brightness'], ax=ax[1], method="ols")
```

[ ]:





[ ]:
```python
acf(pulsar['Brightness'], nlags=10)
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\statsmodels\tsa\stattools.py:667:
FutureWarning: fft=True will become the default after the release of the 0.12
release of statsmodels. To suppress this warning, explicitly set fft=False.
  warnings.warn(
```

[ ]:
```
array([1.        , 0.37138454, 0.34994166, 0.31194031, 0.28665069,
       0.29048719, 0.25431929, 0.27167022, 0.22662943, 0.24809334,
       0.25146666])
```

[ ]:
```python
acfpulsar = pd.DataFrame()
for lag in range(0,11):
    acfpulsar[f"B_lag_{lag}"] = pulsar['Brightness'].shift(lag)


acfpulsar
```

[ ]:
```
       B_lag_0   B_lag_1   B_lag_2   B_lag_3   B_lag_4   B_lag_5   B_lag_6  \
0     0.101127       NaN       NaN       NaN       NaN       NaN       NaN
1     0.012166  0.101127       NaN       NaN       NaN       NaN       NaN
2     0.021918  0.012166  0.101127       NaN       NaN       NaN       NaN
3     0.181179  0.021918  0.012166  0.101127       NaN       NaN       NaN
4     0.000240  0.181179  0.021918  0.012166  0.101127       NaN       NaN
…          …         …         …         …         …         …         …
1814  0.105178  0.008539  0.053246  0.024587  0.004085  0.000947  0.044895
1815  0.064272  0.105178  0.008539  0.053246  0.024587  0.004085  0.000947
1816  0.000171  0.064272  0.105178  0.008539  0.053246  0.024587  0.004085
1817 -0.000924  0.000171  0.064272  0.105178  0.008539  0.053246  0.024587
1818  0.000001 -0.000924  0.000171  0.064272  0.105178  0.008539  0.053246

       B_lag_7   B_lag_8   B_lag_9  B_lag_10
0          NaN       NaN       NaN       NaN
1          NaN       NaN       NaN       NaN
2          NaN       NaN       NaN       NaN
3          NaN       NaN       NaN       NaN
4          NaN       NaN       NaN       NaN
…          …         …         …         …
1814  0.007906  0.048652  0.013009  0.006294
1815  0.044895  0.007906  0.048652  0.013009
1816  0.000947  0.044895  0.007906  0.048652
1817  0.004085  0.000947  0.044895  0.007906
1818  0.024587  0.004085  0.000947  0.044895

[1819 rows x 11 columns]
```

[ ]:
```python
acfpulsar.corr()["B_lag_0"].values
```

[ ]:
```
array([1.        , 0.37158343, 0.35041747, 0.31258703, 0.28752434,
       0.29153195, 0.25533259, 0.27276504, 0.22759855, 0.2492633 ,
```

```
        0.25277541])
```

6.0.1 Getting every 5th as per the auto correlation

6.0.2 Creating a new set of discrete 100 sets and examining them specifically

6.0.3 Further Random testing to move into extensive testing

Getting every 5th as per the auto correlation

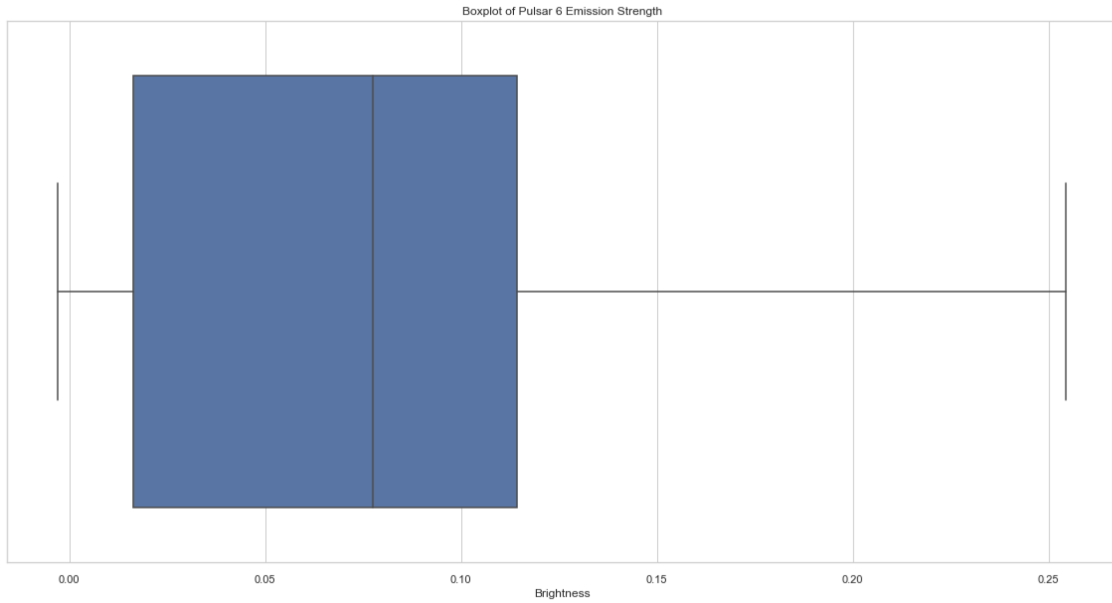
[ ]:
```
held5ths = pulsar[pulsar.index % 5 == 0]
held5ths
```

[ ]:
```
      Pulse Number  Brightness  Uncertainty  Binary
0                1    0.101127     0.001893       1
5                6    0.085866     0.001723       1
10              11    0.123529     0.002026       1
15              16    0.029203     0.001918       0
20              21    0.042757     0.001891       0
…             …          …          …       …
1795          1796    0.004570     0.001779       0
1800          1801    0.002429     0.001749       0
1805          1806    0.013009     0.001764       0
1810          1811    0.004085     0.001713       0
1815          1816    0.064272     0.001995       0

[364 rows x 4 columns]
```

[ ]:
```
medianheld5ths = held5ths["Brightness"].median()
medianheld5ths
```

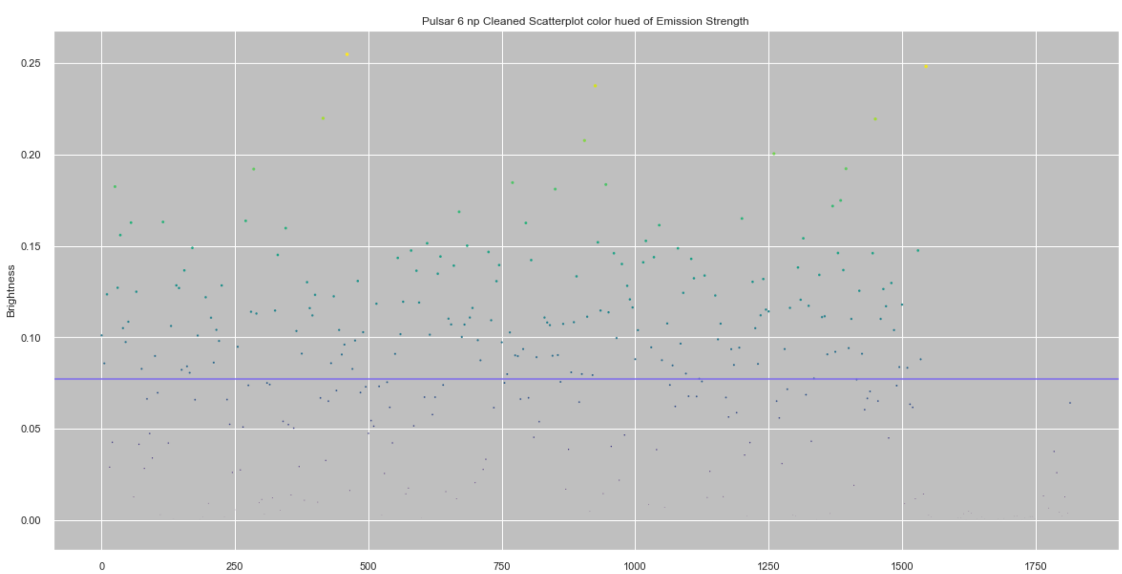
[ ]:
```
0.07756883
```

[ ]:
```
plt.figure(figsize=(20,10))
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=held5ths["Brightness"]).set_title("Boxplot of Pulsar 6␣
 ↪Emission Strength")
```

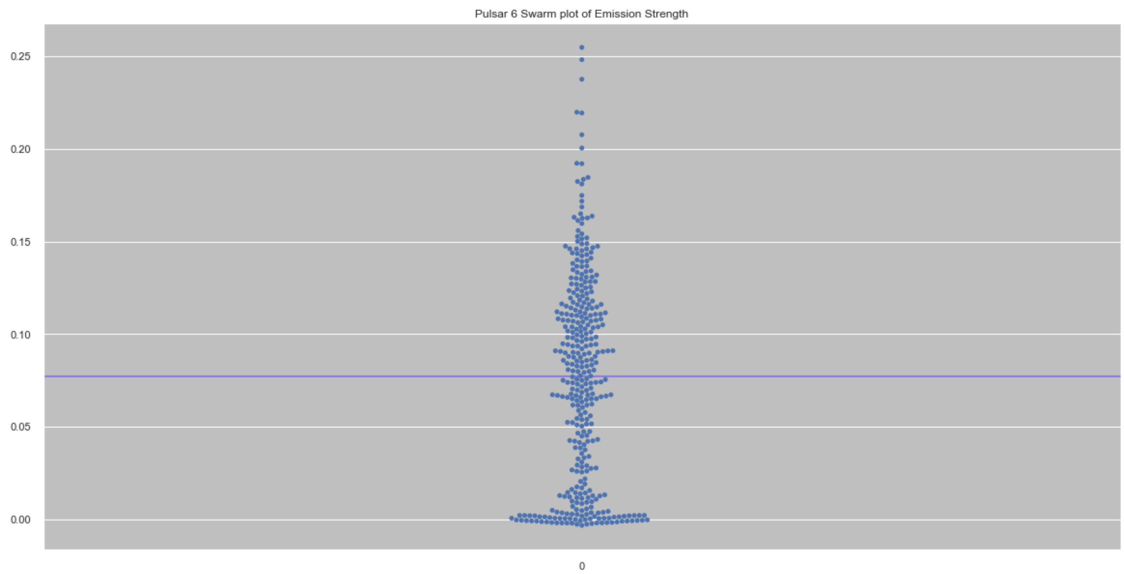


[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = held5ths.Brightness.values
ax = sns.scatterplot(data=held5ths["Brightness"], s= strength*50, c=strength,␣
 ↪cmap="viridis", marker="o").set_title('Pulsar 6 np Cleaned Scatterplot color␣
 ↪hued of Emission Strength')
ax = plt.axhline( y=0.07756883, ls='-',c='mediumslateblue')
```

```
c:\Users\oxlay\anaconda3\lib\site-packages\matplotlib\collections.py:1003:
RuntimeWarning: invalid value encountered in sqrt
  scale = np.sqrt(self._sizes) * dpi / 72.0 * self._factor
```



[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = held5ths.Brightness.values
ax = plt.axhline( y=0.07756883, ls='-',c='mediumslateblue')
ax = sns.swarmplot(data=held5ths["Brightness"], c="blue").set_title('Pulsar 6␣
↪Swarm plot of Emission Strength')
```



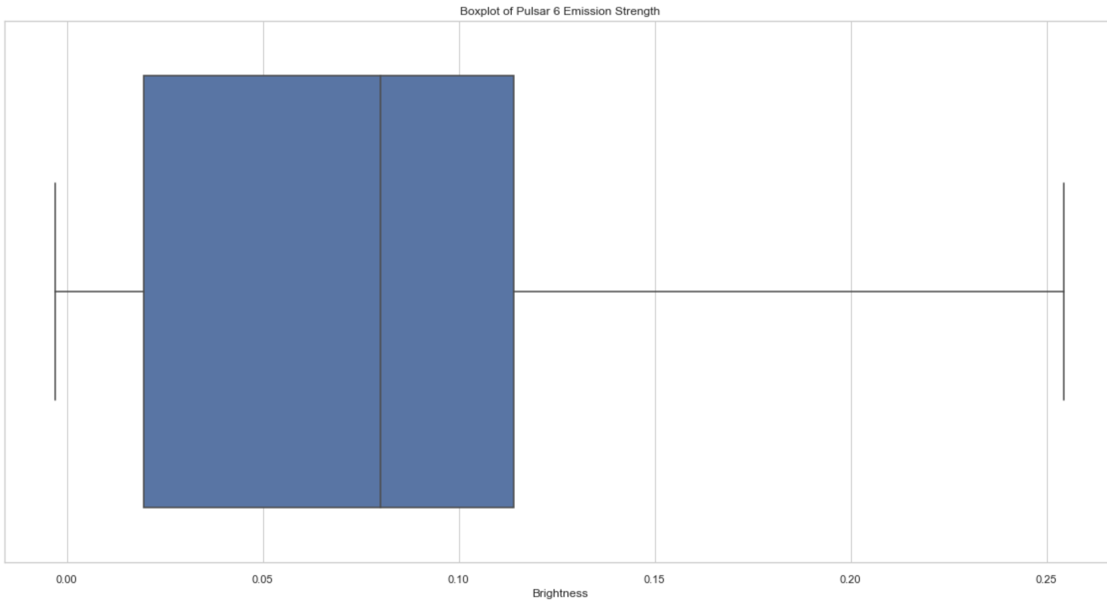
[ ]:
```
print(len(held5ths[(held5ths.Brightness > 0.07756883)]))
print(len(held5ths[(held5ths.Brightness < 0.07756883)]))
```

```
182
182
```

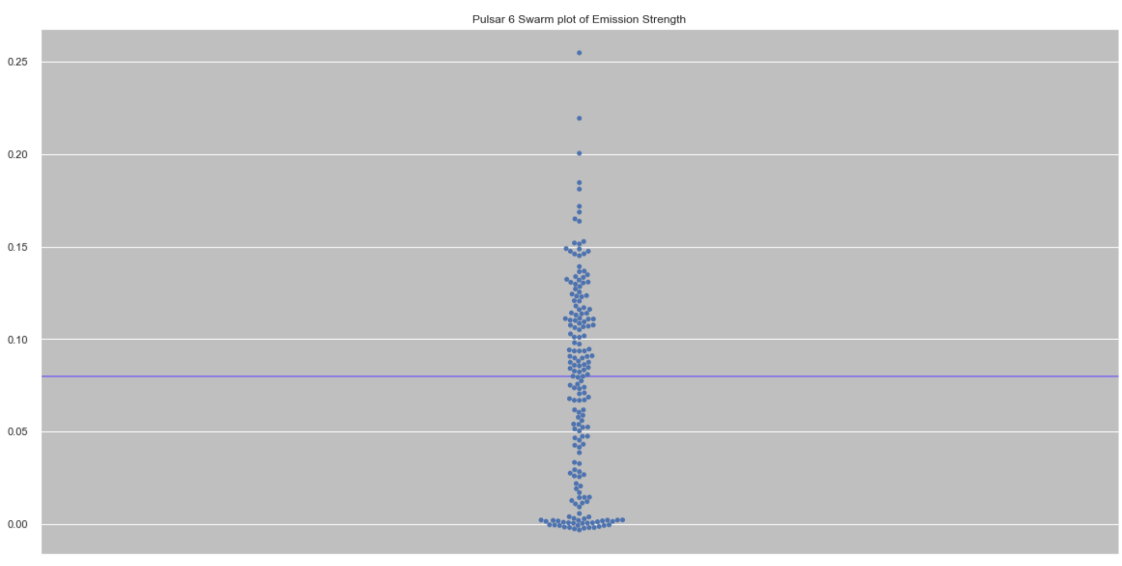
[ ]:
```
held10ths = pulsar[pulsar.index % 10 == 0]
medianheld10ths = held10ths["Brightness"].median()
medianheld10ths
```

[ ]: 0.079977185

[ ]:
```
plt.figure(figsize=(20,10))
sns.set_theme(style="whitegrid")
ax = sns.boxplot(x=held10ths["Brightness"]).set_title("Boxplot of Pulsar 6␣
↪Emission Strength")
```



[ ]:
```
plt.figure(figsize=(20,10))
sns.set_style("darkgrid", {"axes.facecolor": ".75"})
strength = held5ths.Brightness.values
ax = plt.axhline( y=0.079977185, ls='-',c='mediumslateblue')
ax = sns.swarmplot(data=held10ths["Brightness"], c="blue").set_title('Pulsar 6␣
↪Swarm plot of Emission Strength')
```



[ ]:
```
print(len(held10ths[(held10ths.Brightness > 0.079977185)]))
print(len(held10ths[(held10ths.Brightness < 0.079977185)]))
```

```
91
91
```

Randomness testing

[ ]:
```
np.savetxt(r'every5thbinarypulsar4.txt', held5ths.Binary, fmt='%d',␣
↪delimiter='')
np.savetxt(r'allpulsar4.txt', pulsar.Binary, fmt='%d', delimiter='')
np.savetxt(r'every10thbinarypulsar4.txt', held10ths.Binary, fmt='%d',␣
↪delimiter='')
```

[ ]:
```
pulsar.Binary
```

[ ]:
```
0       1
1       0
2       0
3       1
4       0
       ..
1814    1
1815    0
1816    0
1817    0
1818    0
Name: Binary, Length: 1819, dtype: int32
```