ANALYSIS ON MARINE PLASTIC POLLUTION

KIM CHAN YUNG [3035299222]

CONTENTS

- INTRODUCTION
- REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION
- LSTM ANALYSIS ON MARINE PLASTIC POLLUTION
- CONCLUSION

INTRODUCTION

WHAT DO YOU FEEL ?

WHAT DO I FEEL ?

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

OVERVIEW

- Is it possible to predict marine plastic pollution in the future ?
  - Linear Regression.
  - Polynomial Regression.
- Increment / Decrement in the number of plastic litters in the ocean ?

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

BRIEF DISCUSSION OF THE RAW DATA FROM ARCGIS HUB

- First 20 rows of raw data (EXCEL format).

Date	Latitude	Longitude	CD1 (/km^2)	CD2 (/km^2)	CD3 (/km^2)	CD4 (/km^2)	WD1 (g/km^2)	WD2 (g/km^2)	WD3 (g/km^2)	WD4 (g/km^2)	Sea State (Beaufort Scale)	Source	Info
9/1/2010	19.9432	-64.5649	58,102.96	21,259.89	2,226.17		4.45	26.83	4.23		2.50	M. Eriksen	NAG10-SM001
9/1/2010	20.2173	-64.3828	6,639.79	4,031.30	1,067.11		1.04	28.69	40.79		2.00	M. Eriksen	NAG10-SM002
9/1/2010	20.4521	-64.1968	15,246.71	12,147.79	991.66		2.57	42.62	3,503.27		2.00	M. Eriksen	NAG10-SM003
10/1/2010	21.1293	-63.8333	5,347.35	6,851.29	1,420.39		1.15	12.86	4.26		2.00	M. Eriksen	NAG10-SM004
10/1/2010	21.4730	-63.5899	4,090.58	5,317.76	409.06		0.48	29.63	4.36		2.00	M. Eriksen	NAG10-SM005
10/1/2010	21.7367	-63.4227	44,914.59	46,144.76	33,133.71		6.26	47.98	2,051.84		2.00	M. Eriksen	NAG10-SM006
11/1/2010	22.1500	-63.1474	17,324.82	4,519.52	502.17		1.38	3.52	1.13		2.00	M. Eriksen	NAG10-SM007
11/1/2010	22.8418	-62.7592	10,467.48	10,329.75	413.19		1.24	33.90	15.84		2.00	M. Eriksen	NAG10-SM008
12/1/2010	24.3576	-62.8129	7,434.53	9,436.14	190.63		1.14	28.59	0.29		4.00	M. Eriksen	NAG10-SM009
12/1/2010	24.3576	-62.8129	0.00	1,510.37	704.84		0.00	8.96	13.59		4.00	M. Eriksen	NAG10-SM010
13/01/2010	24.7657	-62.7221	377.59	1,636.23	377.59		0.11	5.41	7.93		4.00	M. Eriksen	NAG10-SM011
13/01/2010	25.3751	-62.4785	3,126.56	2,918.12	416.87		0.46	3.99	0.42		3.00	M. Eriksen	NAG10-SM012
13/01/2010	25.5607	-62.4387	428,395.48	132,757.97	119,370.62		15.62	837.83	25,752.82		2.00	M. Eriksen	NAG10-SM013
14/01/2010	26.0914	-61.7168	0.00	2,111.27	351.88		0.00	4.52	8.45		4.50	M. Eriksen	NAG10-SM014
15/01/2010	29.2490	-62.9857	4,682.38	14,983.63	1,560.79		0.61	86.00	105.82		2.50	M. Eriksen	NAG10-SM015
15/01/2010	29.7453	-63.2167	3,020.74	35,544.01	3,725.58		0.41	165.27	613.71		2.00	M. Eriksen	NAG10-SM016
16/01/2010	30.5477	-63.7064	31,985.11	157,152.30	4,991.90		8.10	778.18	599.95		2.00	M. Eriksen	NAG10-SM017
17/01/2010	31.1813	-64.0147	1,173.82	10,137.53	1,814.08		0.45	47.17	101.16		2.50	M. Eriksen	NAG10-SM018
17/01/2010	31.5350	-64.2677	9,953.75	31,224.77	2,590.70		1.50	135.67	67.49		2.00	M. Eriksen	NAG10-SM019
18/01/2010	31.9644	-64.5448	4,082.42	24,984.39	2,776.04		0.65	154.97	1,465.26		2.00	M. Eriksen	NAG10-SM020
28/01/2010	32.3200	-64.4322	6,826.00	37,944.54	4,818.35		1.20	287.29	307.97		2.00	M. Eriksen	NAG10-SM021
28/01/2010	32.0790	-64.6316	0.00	52,359.45	3,272.47		0.00	449.15	175.08		2.50	M. Eriksen	NAG10-SM022

https://hub.arcgis.com/datasets/CESJ::estimate-of-plastic-pollution-in-the-worlds-oceans-km2-4-76-200-mm/data?selectedAttribute=WD3__G_KM_

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

BRIEF DISCUSSION OF THE RAW DATA FROM ARCGIS HUB

- CD is the count density of the plastic (/ km^2).
- WD is the weight density of the plastic (g / km^2).

CD1_(/km^2) 0.335-0.999 mm	WD1_(g/km^2) 0.335-0.999 mm
CD2_(/km^2) 1.00-4.75 mm	WD2_(g/km^2) 1.00-4.75 mm
CD3_(/km^2) 4.75-200 mm	WD3_(g/km^2) 4.75-200 mm
CD4_(/km^2) >200 mm	WD4_(g/km^2) >200 mm

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

IMPORT THE DATA

```
import pandas as pd

from google.colab import drive

drive.mount('/content/gdrive/')
pollution_data = pd.read_csv('/content/gdrive/My Drive/AI_NoteBook/PROJECT/Import_File/PlasticMarinePollutionDataset.csv')
```

```
Drive already mounted at /content/gdrive/; to attempt to forcibly remount, call drive.mount("/content/gdrive/", force_remount=True).
```

	Date	Latitude	Longitude	CD1_(/km^2)	CD2_(/km^2)	CD3_(/km^2)	CD4 _(/km^2)	WD1_(g/km^2)	WD2_(g/km^2)	WD3_(g/km^2)	WD4_(g/km^2)	Sea State	Source	Info	Comments	Unnamed: 15
0	9/1/2010	19.9432	-64.5649	58,102.96	21,259.89	2,226.17	NaN	4.45	26.83	4.23	NaN	2.5	M. Eriksen	NAG10-SM001	NaN	NaN
1	9/1/2010	20.2173	-64.3828	6,639.79	4,031.30	1,067.11	NaN	1.04	28.69	40.79	NaN	2.0	M. Eriksen	NAG10-SM002	NaN	NaN
2	9/1/2010	20.4521	-64.1968	15,246.71	12,147.79	991.66	NaN	2.57	42.62	3,503.27	NaN	2.0	M. Eriksen	NAG10-SM003	NaN	NaN
3	10/1/2010	21.1293	-63.8333	5,347.35	6,851.29	1,420.39	NaN	1.15	12.86	4.26	NaN	2.0	M. Eriksen	NAG10-SM004	NaN	NaN
4	10/1/2010	21.4730	-63.5899	4,090.58	5,317.76	409.06	NaN	0.48	29.63	4.36	NaN	2.0	M. Eriksen	NAG10-SM005	NaN	NaN
5	10/1/2010	21.7367	-63.4227	44,914.59	46,144.76	33,133.71	NaN	6.26	47.98	2,051.84	NaN	2.0	M. Eriksen	NAG10-SM006	NaN	NaN
6	11/1/2010	22.1500	-63.1474	17,324.82	4,519.52	502.17	NaN	1.38	3.52	1.13	NaN	2.0	M. Eriksen	NAG10-SM007	NaN	NaN
7	11/1/2010	22.8418	-62.7592	10,467.48	10,329.75	413.19	NaN	1.24	33.9	15.84	NaN	2.0	M. Eriksen	NAG10-SM008	NaN	NaN
8	12/1/2010	24.3576	-62.8129	7,434.53	9,436.14	190.63	NaN	1.14	28.59	0.29	NaN	4.0	M. Eriksen	NAG10-SM009	NaN	NaN
9	12/1/2010	24.3576	-62.8129	0	1,510.37	704.84	NaN	0.00	8.96	13.59	NaN	4.0	M. Eriksen	NAG10-SM010	NaN	NaN

```
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Date            1571 non-null   object
 1   Latitude        1571 non-null   float64
 2   Longitude       1571 non-null   float64
 3   CD1_(/km^2)     679 non-null    object
 4   CD2_(/km^2)     679 non-null    object
 5   CD3_(/km^2)     807 non-null    object
 6   CD4 _(/km^2)    1089 non-null   float64
 7   WD1_(g/km^2)    441 non-null    float64
 8   WD2_(g/km^2)    441 non-null    object
 9   WD3_(g/km^2)    569 non-null    object
 10  WD4_(g/km^2)    887 non-null    object
 11  Sea State       1213 non-null   float64
 12  Source          1571 non-null   object
 13  Info            1571 non-null   object
 14  Comments        37 non-null     object
 15  Unnamed: 15     377 non-null    object
dtypes: float64(5), object(11)
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

MODIFICATION OF THE RAW DATA

- Conversion of EXCEL file into CSV file format.
- Renaming of the columns.

```
pollution_data.rename(columns = {'CD1_(/km^2)' : 'Count_Density_Class1Plastic'}, inplace = True)
pollution_data.rename(columns = {'CD2_(/km^2)' : 'Count_Density_Class2Plastic'}, inplace = True)
pollution_data.rename(columns = {'CD3_(/km^2)' : 'Count_Density_Class3Plastic'}, inplace = True)
pollution_data.rename(columns = {'CD4 _(/km^2)' : 'Count_Density_Class4Plastic'}, inplace = True)
pollution_data.rename(columns = {'WD1_(g/km^2)' : 'Weight_Density_Class1Plastic'}, inplace = True)
pollution_data.rename(columns = {'WD2_(g/km^2)' : 'Weight_Density_Class2Plastic'}, inplace = True)
pollution_data.rename(columns = {'WD3_(g/km^2)' : 'Weight_Density_Class3Plastic'}, inplace = True)
pollution_data.rename(columns = {'WD4_(g/km^2)' : 'Weight_Density_Class4Plastic'}, inplace = True)
```

- Conversion of datatype on density (CD1, CD2, …) columns.

```
pollution_data['Count_Density_Class1Plastic'] = pollution_data.Count_Density_Class1Plastic.str.replace(',', '').astype(float)
pollution_data['Count_Density_Class2Plastic'] = pollution_data.Count_Density_Class2Plastic.str.replace(',', '').astype(float)
pollution_data['Count_Density_Class3Plastic'] = pollution_data.Count_Density_Class3Plastic.str.replace(',', '').astype(float)
pollution_data['Weight_Density_Class2Plastic'] = pollution_data.Weight_Density_Class2Plastic.str.replace(',', '').astype(float)
pollution_data['Weight_Density_Class3Plastic'] = pollution_data.Weight_Density_Class3Plastic.str.replace(',', '').astype(float)
pollution_data['Weight_Density_Class4Plastic'] = pollution_data.Weight_Density_Class4Plastic.str.replace(',', '').astype(float)
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

PREPROCESSING

- Remove duplicates.
- Remove nulls.
- Figuring out the important features.
  - Density columns.
  - Date column.

```
pollution_data = pollution_data.drop_duplicates(keep = 'first')
```

```
pollution_data_1 = pollution_data[['Date','Count_Density_Class1Plastic']]
```

```
pollution_data_1 = pollution_data_1.dropna(how = 'any')
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

PREPROCESSING

- Conversion of datatype of the ‘Date' column.
- Find the total density for each date.
  - Group the rows of the same Date.
  - Sum up the density.

```
pollution_data_1['Date'] = pd.to_datetime(pollution_data_1.Date)

pollution_data_1_GroupByDate = pollution_data_1.groupby('Date').Count_Density_Class1Plastic.sum()

pollution_data_1_GroupByDate = pd.DataFrame(pollution_data_1_GroupByDate)

pollution_data_1_GroupByDate ['Date'] = pollution_data_1_GroupByDate.index
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

PREPROCESSING

- Define a column of how many days from the starting date.

```
startdate = pd.to_datetime('2007/09/16')

pollution_data_1_GroupByDate ['Days after 2007-09-16'] = (pollution_data_1_GroupByDate.Date).subtract(startdate)

pollution_data_1_GroupByDate ['Days after 2007-09-16']  = pollution_data_1_GroupByDate['Days after 2007-09-16'].astype(str)

pollution_data_1_GroupByDate ['Days after 2007-09-16'] = pollution_data_1_GroupByDate ['Days after 2007-09-16'].str.replace(' days 00:00:00.000000000', '')
```

Date	Count_Density_Class1Plastic	Date
2007-09-16	47979.38	2007-09-16
2007-09-18	118621.69	2007-09-18
2007-09-20	31977.91	2007-09-20
2007-09-21	16235.81	2007-09-21
2007-09-22	102961.97	2007-09-22
2007-09-24	36104.52	2007-09-24

Date	Count_Density_Class1Plastic	Date	Days after 2007-09-16
2007-09-16	47979.38	2007-09-16	0
2007-09-18	118621.69	2007-09-18	2
2007-09-20	31977.91	2007-09-20	4
2007-09-21	16235.81	2007-09-21	5
2007-09-22	102961.97	2007-09-22	6
2007-09-24	36104.52	2007-09-24	8

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

PREPROCESSING

Date	Count_Density_Class2Plastic	Date
2007-09-16	86770.68	2007-09-16
2007-09-18	97000.01	2007-09-18
2007-09-20	45723.96	2007-09-20
2007-09-21	255134.23	2007-09-21

Date	Count_Density_Class2Plastic	Date	Days after 2007-09-16
2007-09-16	86770.68	2007-09-16	0
2007-09-18	97000.01	2007-09-18	2
2007-09-20	45723.96	2007-09-20	4
2007-09-21	255134.23	2007-09-21	5

Date	Count_Density_Class3Plastic	Date
2007-09-16	15456.10	2007-09-16
2007-09-18	14282.21	2007-09-18
2007-09-20	5025.32	2007-09-20
2007-09-21	11597.01	2007-09-21

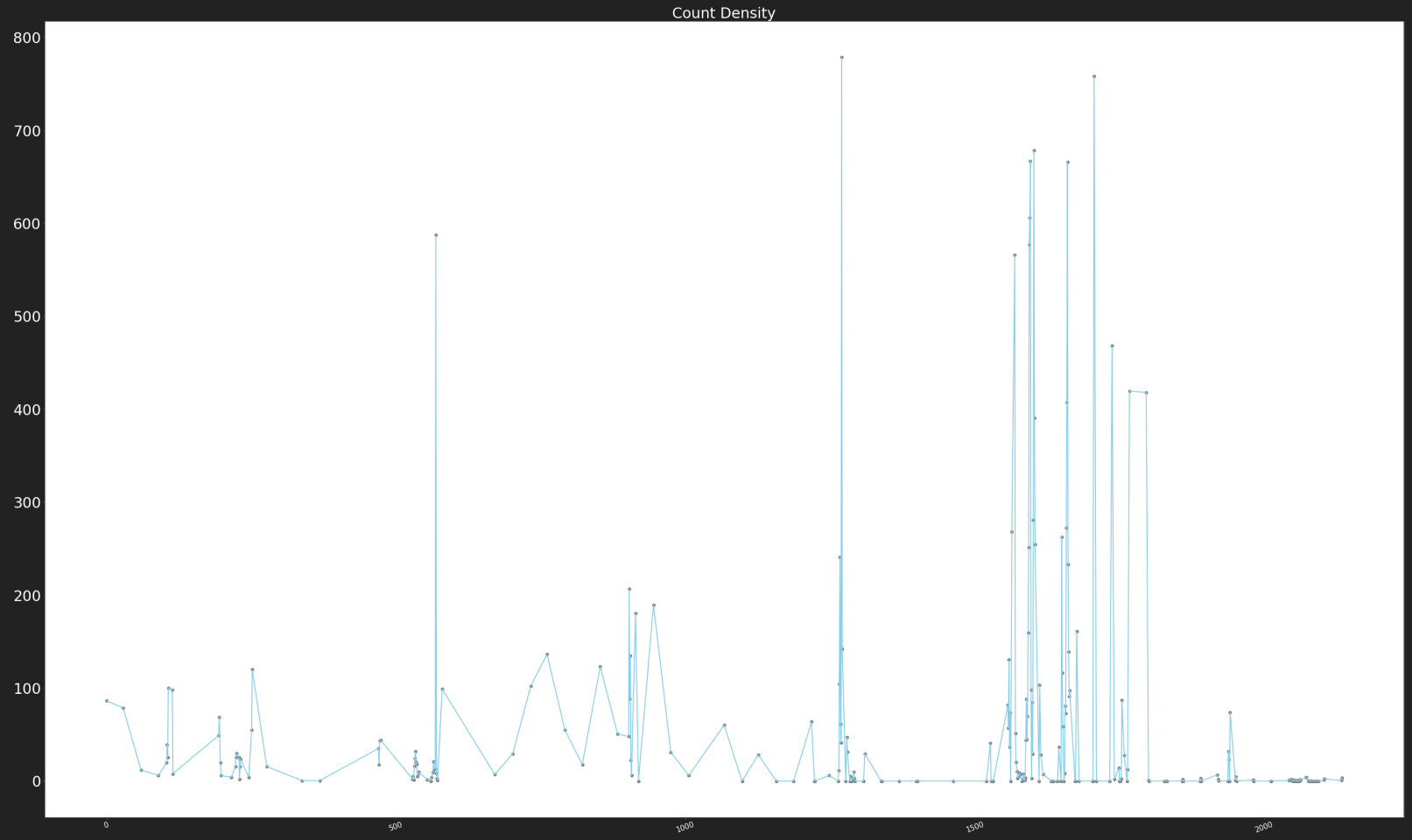
Date	Count_Density_Class3Plastic	Date	Days after 2007-09-16
2007-09-16	15456.10	2007-09-16	0
2007-09-18	14282.21	2007-09-18	2
2007-09-20	5025.32	2007-09-20	4
2007-09-21	11597.01	2007-09-21	5

Date	Count_Density_Class4Plastic	Date
2008-02-06	86.43	2008-02-06
2008-03-06	78.56	2008-03-06
2008-04-06	11.78	2008-04-06
2008-05-06	5.89	2008-05-06

Date	Count_Density_Class4Plastic	Date	Days after 2008-02-06
2008-02-06	86.43	2008-02-06	0
2008-03-06	78.56	2008-03-06	29
2008-04-06	11.78	2008-04-06	60
2008-05-06	5.89	2008-05-06	90

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

PLOT OF COUNT DENSITY OF CLASS 4 PLASTIC



REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

POLYNOMIAL REGRESSION TO PREDICT CLASS 4 PLASTIC

- Split dataset.
- Trial of various degree.
  - Degree = 1.
  - Degree = 2.
  - Degree = 3.
  - Degree = 10.

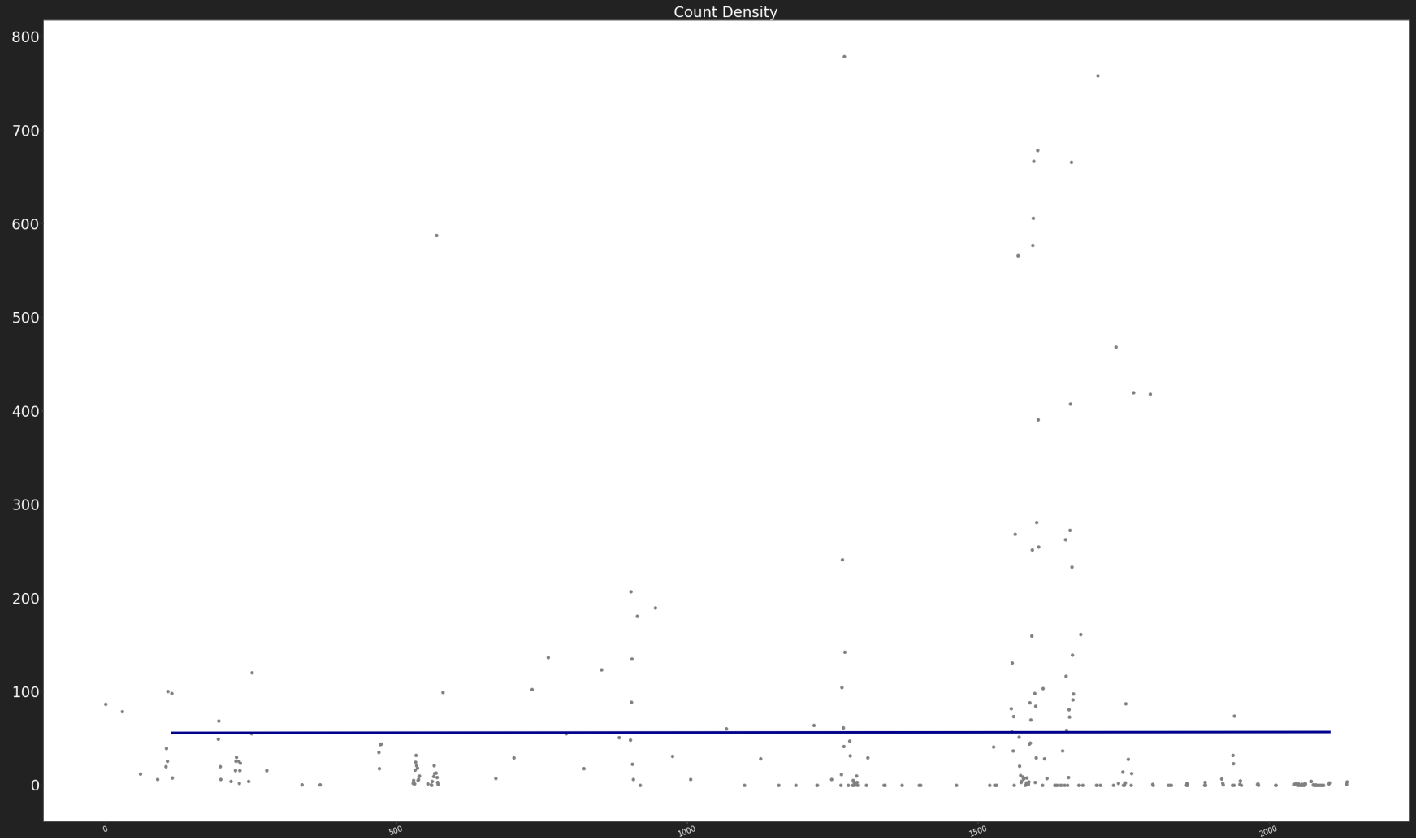
```
########## [ DEFINE MODEL AND TRAIN  < Degree = 1 > ] ##########
polynomial = PolynomialFeatures(degree=1)

X_train_transformed_for_poly = polynomial.fit_transform(X_train)

poly_linear_model = linear_model.LinearRegression()

poly_linear_model.fit(X_train_transformed_for_poly, y_train)
```

```
########## [ DEFINE MODEL AND TRAIN  < Degree = 3 > ] ##########
polynomial = PolynomialFeatures(degree=3)

X_train_transformed_for_poly = polynomial.fit_transform(X_train)

poly_linear_model = linear_model.LinearRegression()

poly_linear_model.fit(X_train_transformed_for_poly, y_train)
```

```
########## [ DEFINE MODEL AND TRAIN  < Degree = 10 > ] ##########
polynomial = PolynomialFeatures(degree=10)

X_train_transformed_for_poly = polynomial.fit_transform(X_train)

poly_linear_model = linear_model.LinearRegression()

poly_linear_model.fit(X_train_transformed_for_poly, y_train)
```

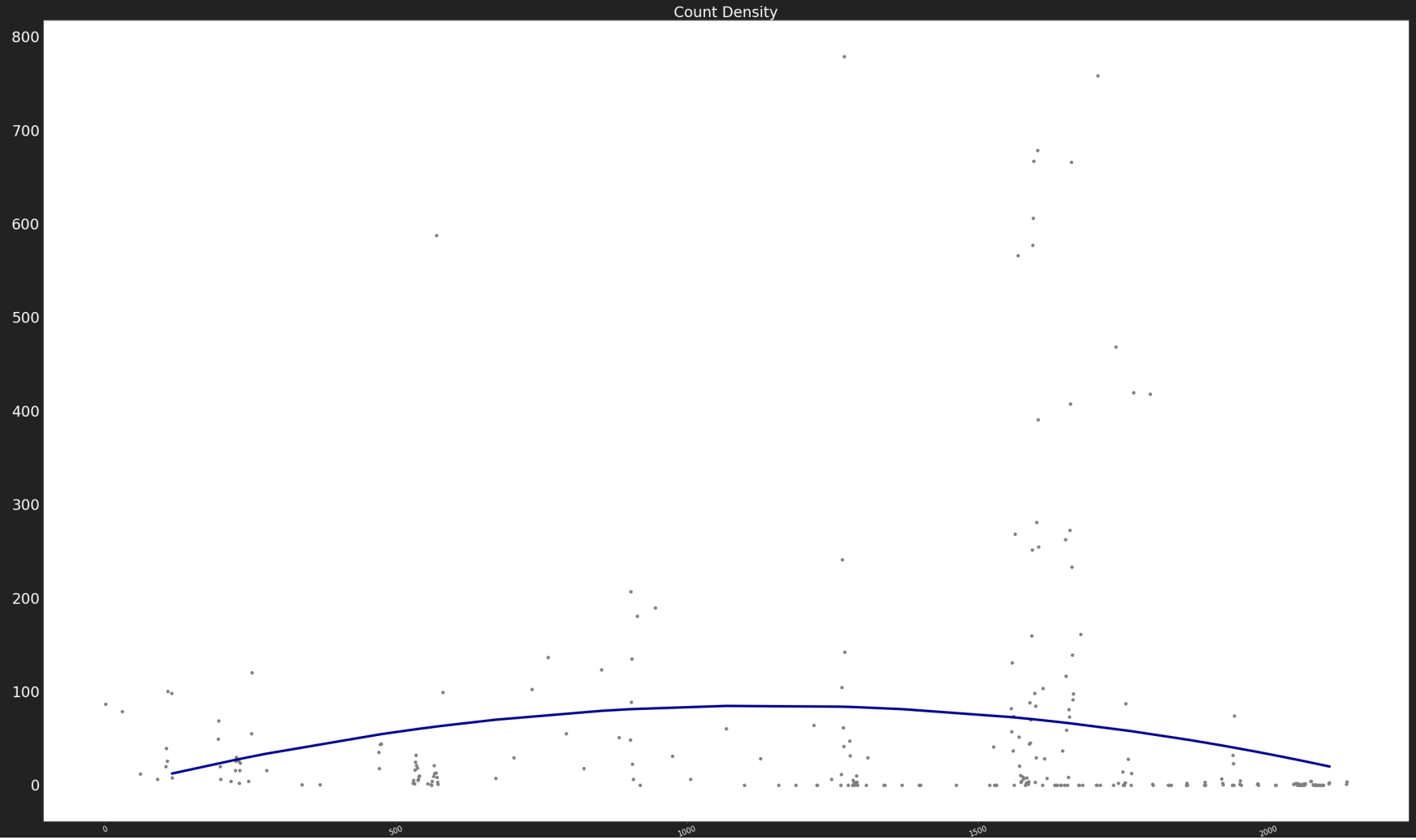
REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

POLYNOMIAL REGRESSION OF CLASS 4 PLASTIC WITH DEGREE = 1



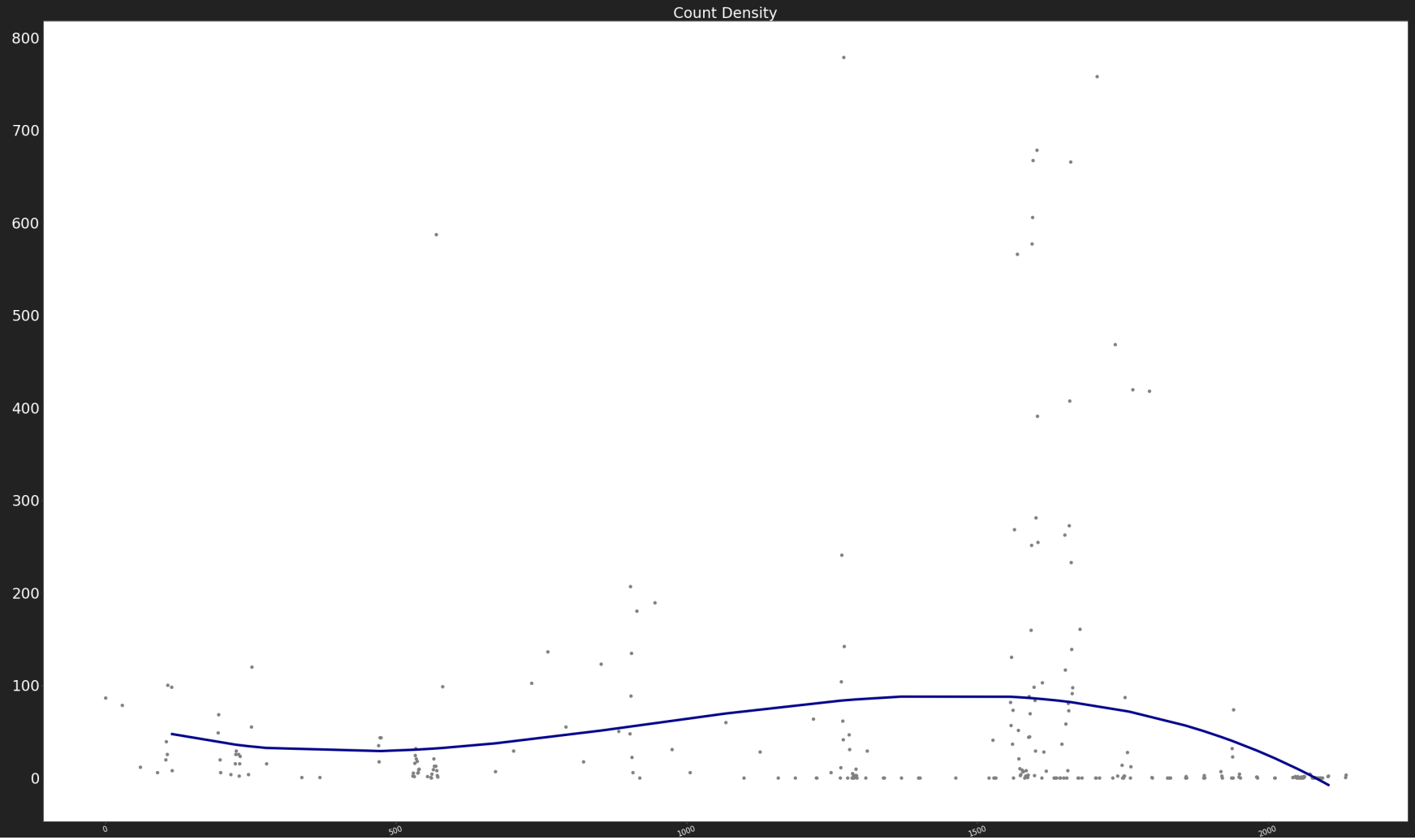
REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

POLYNOMIAL REGRESSION OF CLASS 4 PLASTIC WITH DEGREE = 2



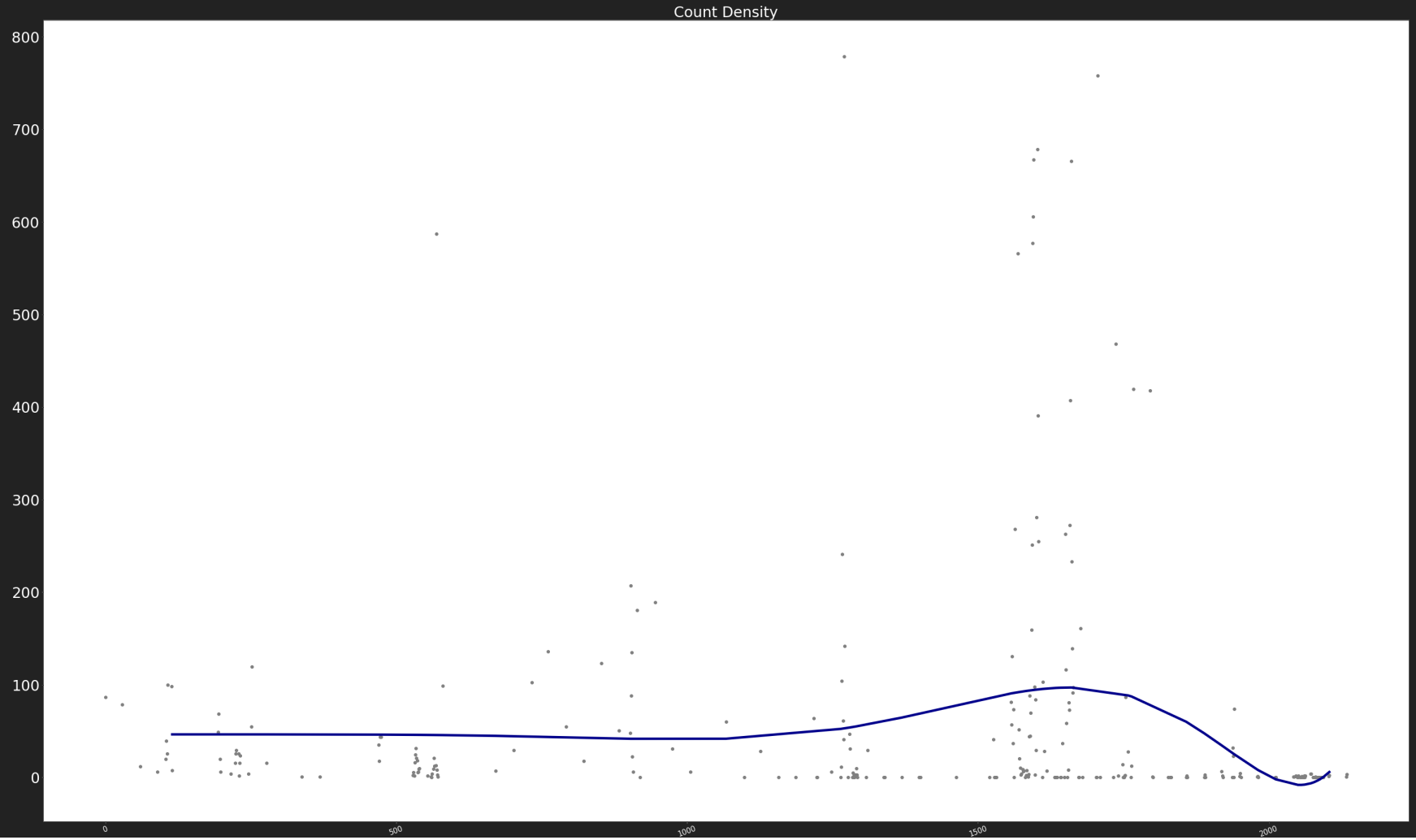
REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

POLYNOMIAL REGRESSION OF CLASS 4 PLASTIC WITH DEGREE = 3



REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

POLYNOMIAL REGRESSION OF CLASS 4 PLASTIC WITH DEGREE = 10



REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

IS THE PREDICTION ACCURATE ?

- NOT ACCURATE AT ALL !!!
- Unclear correlation.
- Uncertain results
- The performance measure is ambiguous to calculate.
  - Mean absolute error.
  - Mean squared error.
  - Median absolute error.

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

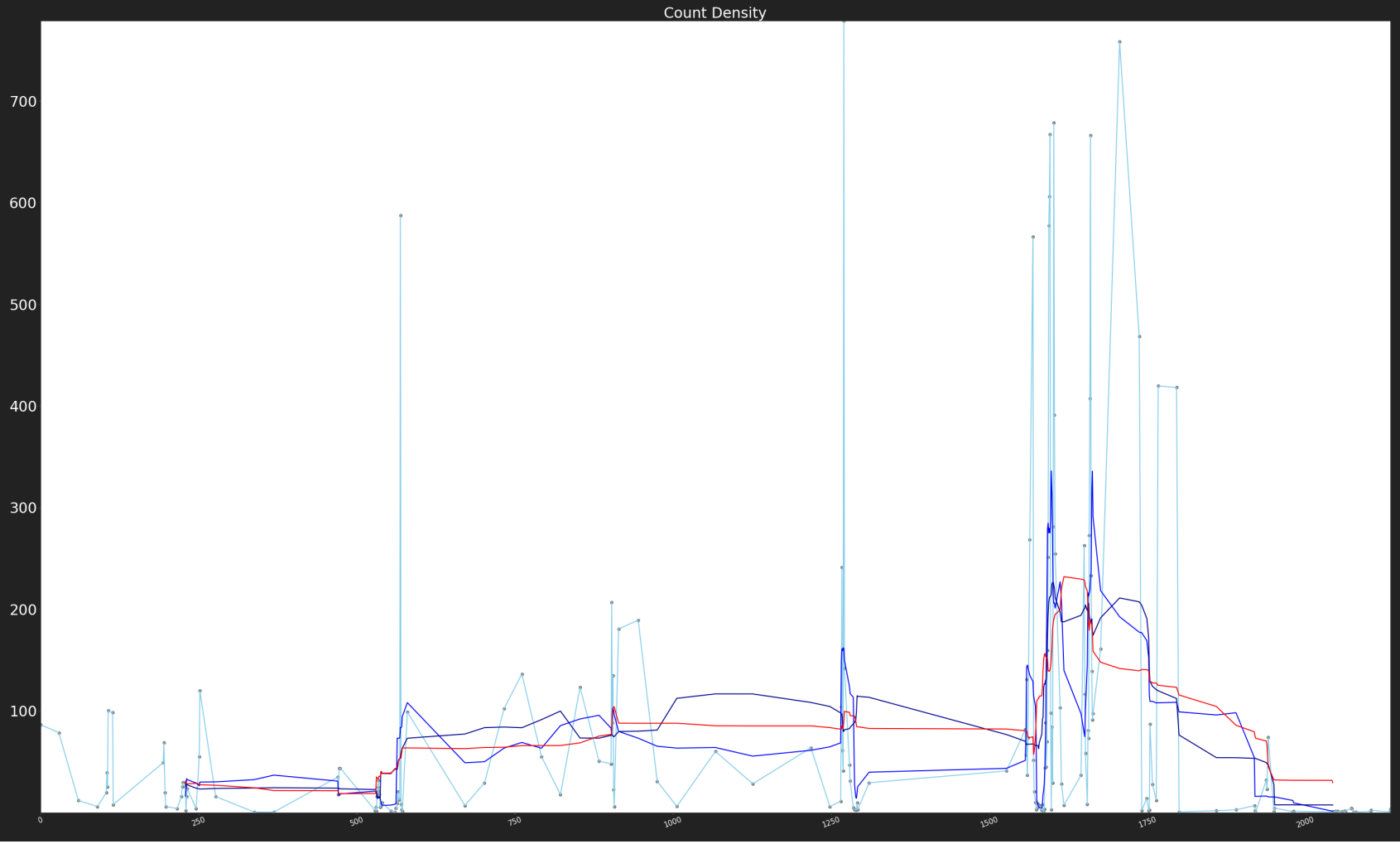
HOW TO PREDICT BETTER ?

- Required better training data.
  - Remove obvious outliers (zeros) from the dataset.
  - ‘Moving Average’ approach to convert noisy to smooth dataset.

```
########## [REMOVE ZEROS] #########
pollution_data_1 = pollution_data_1.drop(index = pollution_data_1[pollution_data_1['Count_Density_Class1Plastic'] == 0].index)

pollution_data_1['Date'] = pd.to_datetime(pollution_data_1.Date)

pollution_data_1_GroupByDate = pollution_data_1.groupby('Date').Count_Density_Class1Plastic.sum()

pollution_data_1_GroupByDate = pd.DataFrame(pollution_data_1_GroupByDate)

pollution_data_1_GroupByDate ['Date'] = pollution_data_1_GroupByDate.index

######### [MOVING AVERAGES APPROACH] #########
pollution_data_1_GroupByDate ['Moving_Average_9'] = pollution_data_1_GroupByDate.Count_Density_Class1Plastic.rolling(9, center=True).mean()
pollution_data_1_GroupByDate ['Moving_Average_21'] = pollution_data_1_GroupByDate.Count_Density_Class1Plastic.rolling(21, center=True).mean()
pollution_data_1_GroupByDate ['Moving_Average_36'] = pollution_data_1_GroupByDate.Count_Density_Class1Plastic.rolling(36, center=True).mean()
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

PLOT OF COUNT DENSITY OF CLASS 4 PLASTIC



REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

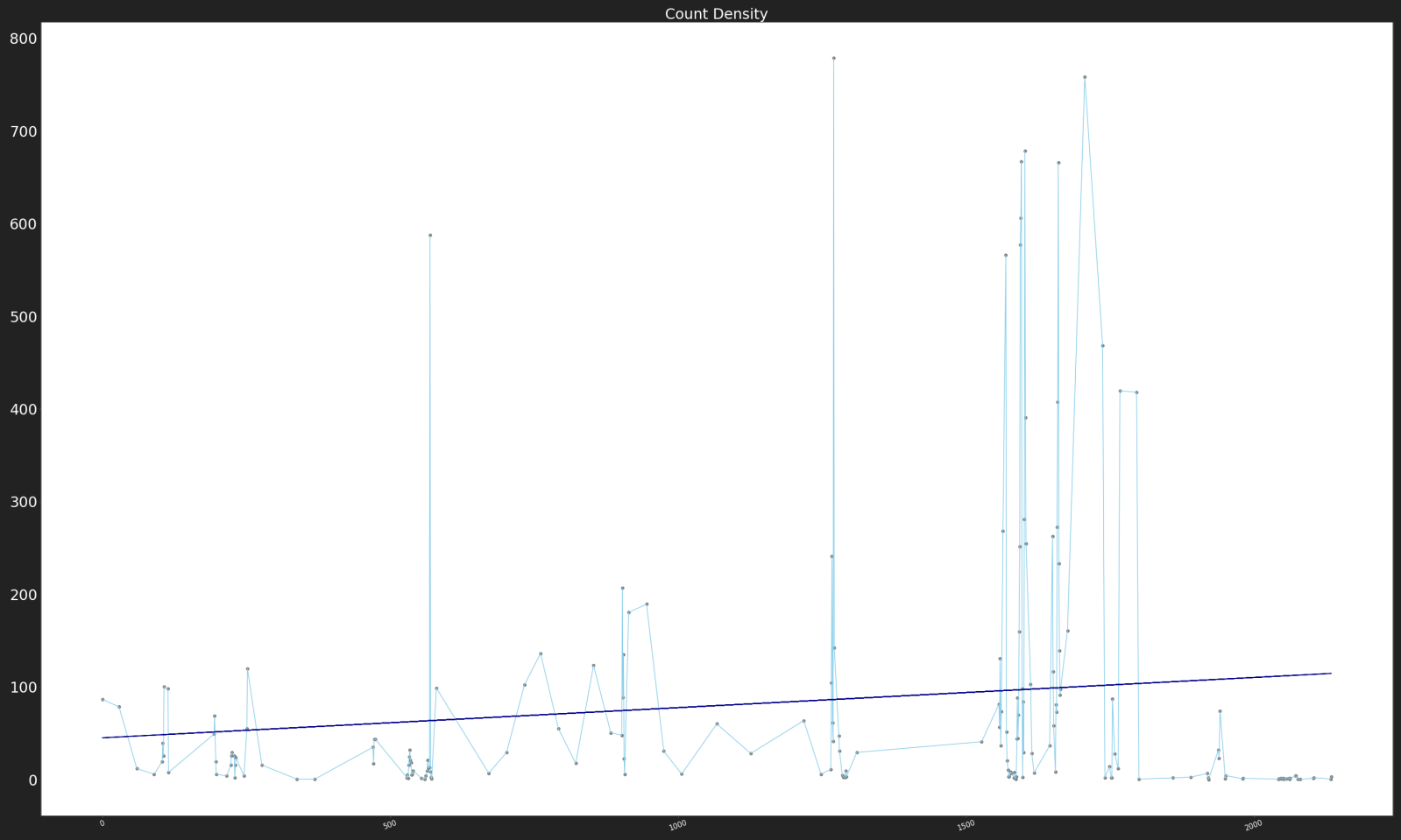
POLYNOMIAL REGRESSION TO PREDICT CLASS 4 PLASTIC

- Trial of various degree.
  - Degree = 1, 2, 3, 4, 5, 6, 7, 8, 10, 30.
- The most accurate result obtained from linear regression (Degree = 1).

```
######### [ DEFINE MODEL AND TRAIN  < Degree = 1 > ] #########
polynomial = PolynomialFeatures(degree=1)

X_train_transformed_for_poly = polynomial.fit_transform(X_train)

poly_linear_model = linear_model.LinearRegression()

poly_linear_model.fit(X_train_transformed_for_poly, y_train)
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

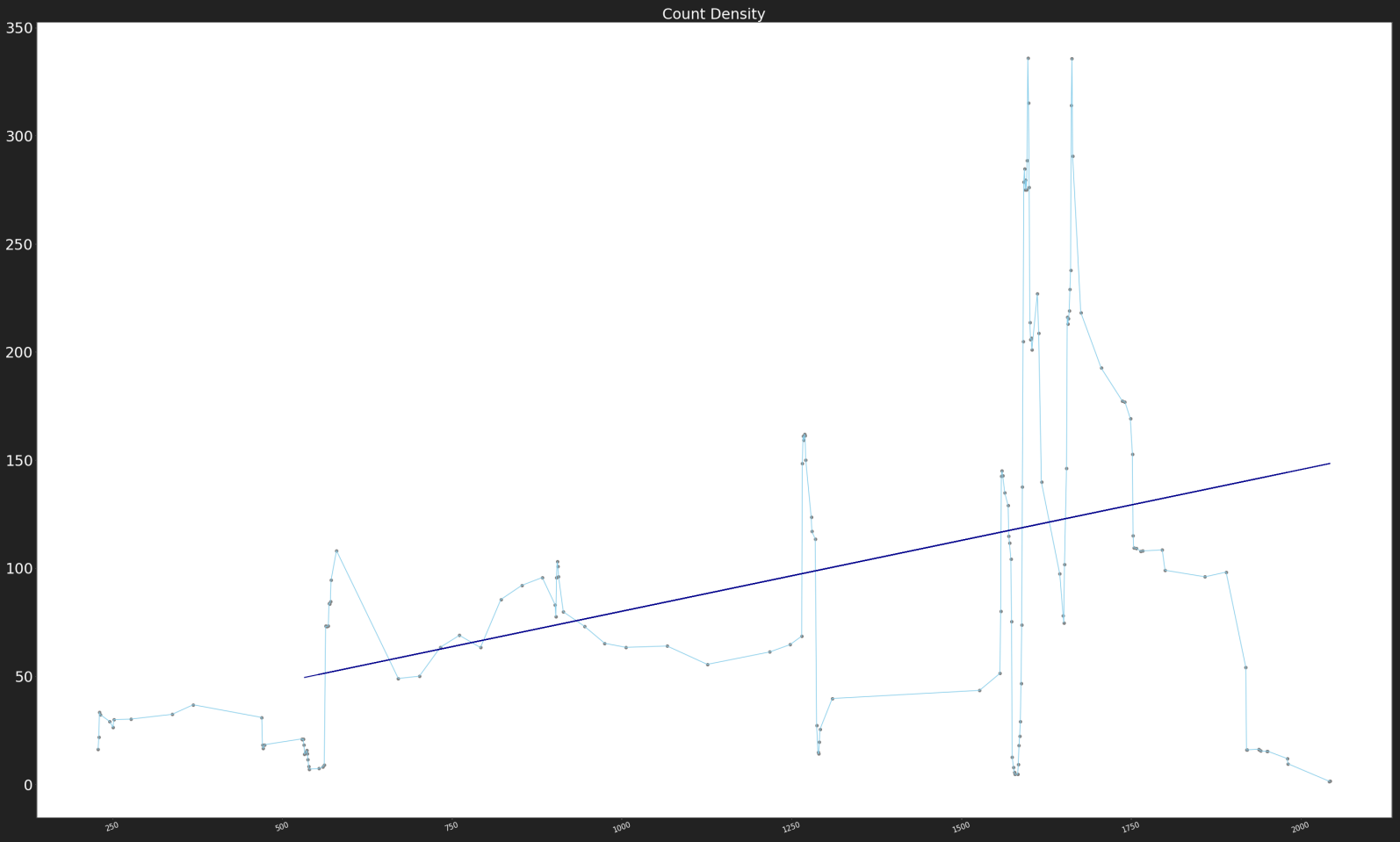
LINEAR REGRESSION OF CLASS 4 PLASTIC.




```
Linear Regressor performance:
Mean absolute error =  82.59
Mean squared error =  15152.65
Median absolute error =  55.56
Explained variance score =  0.03
R2 score =  0.03
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

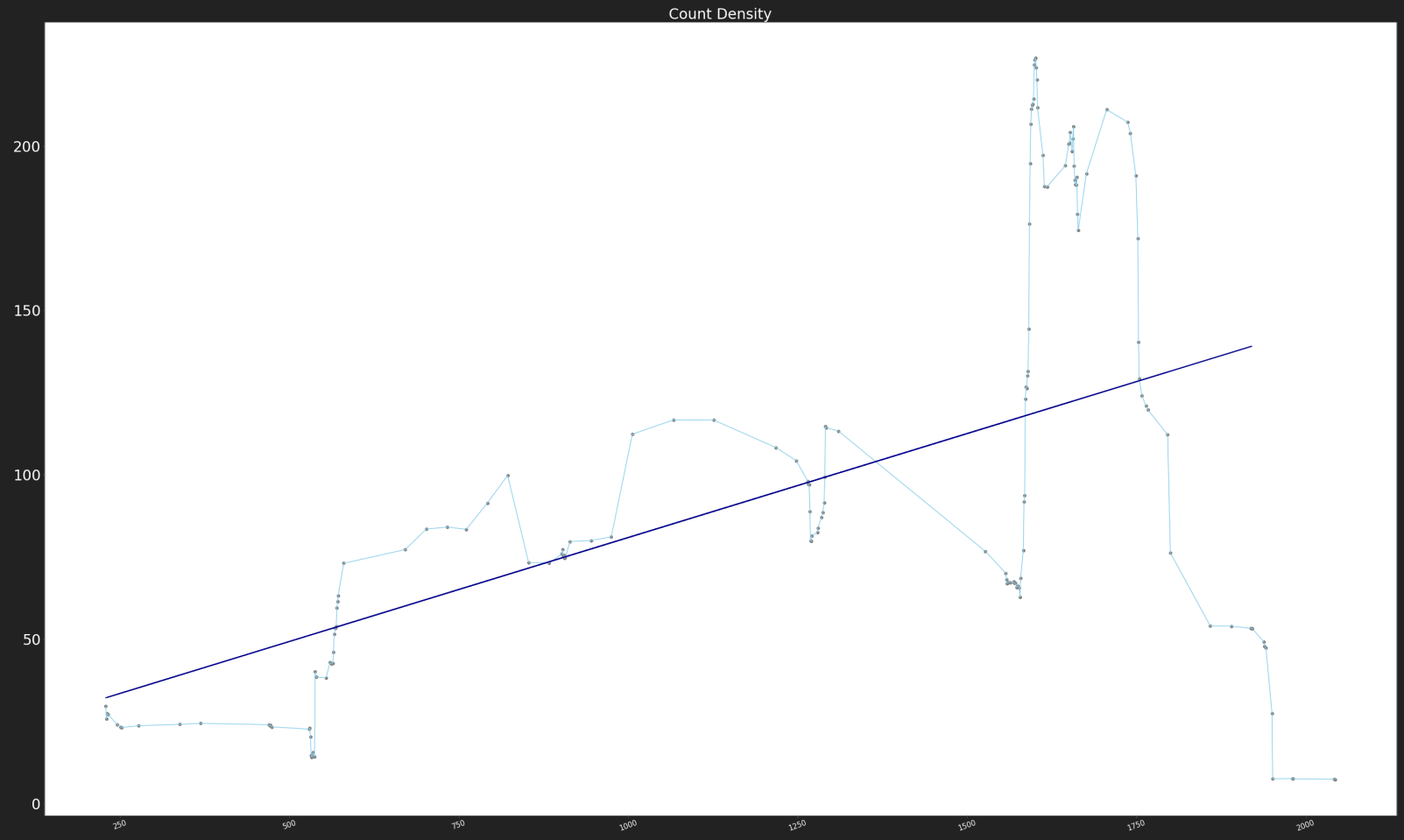
LINEAR REGRESSION OF CLASS 4 PLASTIC (9 MOVING AVERAGE).




```
Linear Regressor performance:
Mean absolute error =  66.12
Mean squared error =  7120.79
Median absolute error =  49.96
Explained variance score =  0.08
R2 score =  0.08
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

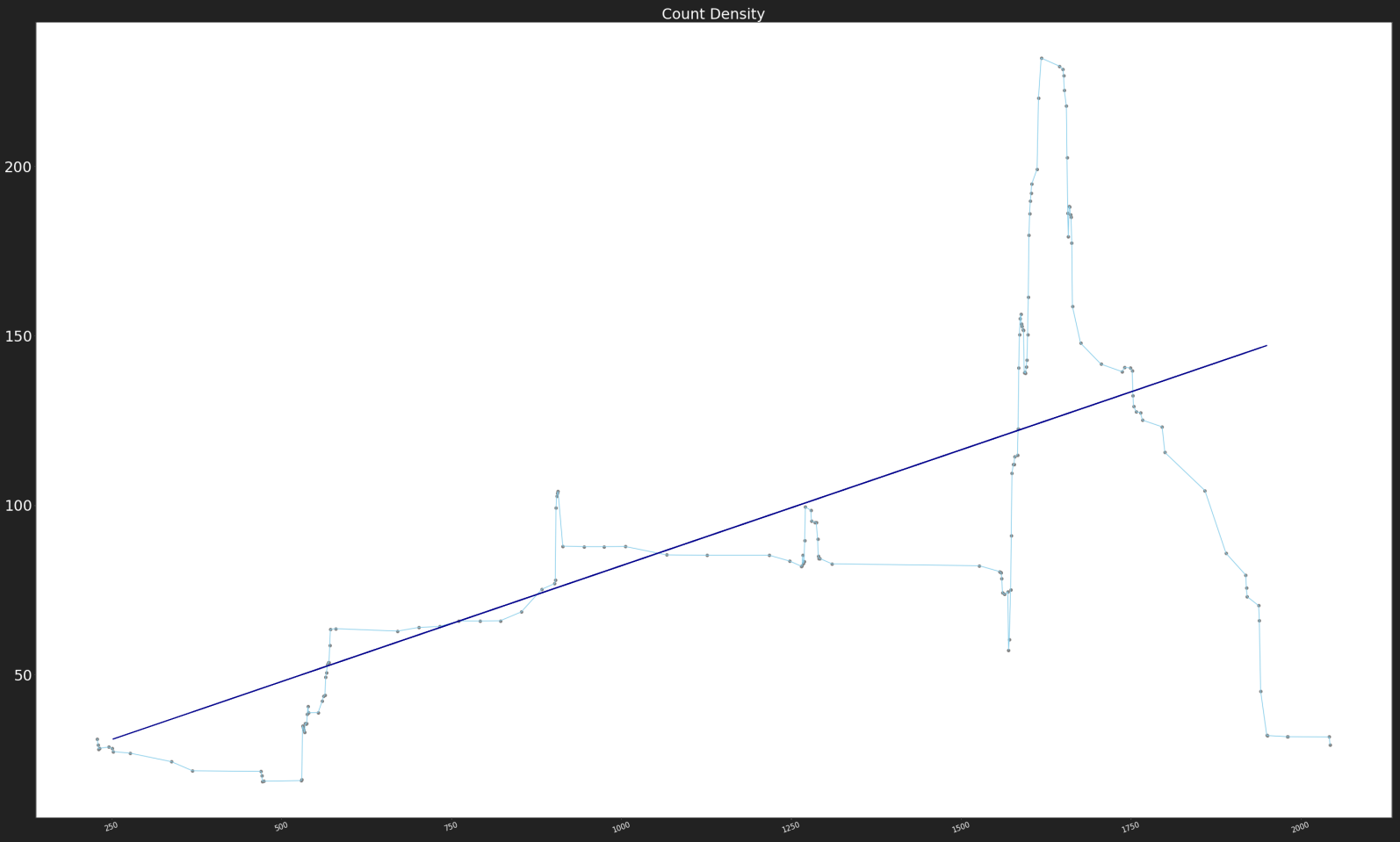
LINEAR REGRESSION OF CLASS 4 PLASTIC (21 MOVING AVERAGE).




```
Linear Regressor performance:
Mean absolute error =  28.01
Mean squared error =  1525.72
Median absolute error =  16.27
Explained variance score =  0.5
R2 score =  0.5
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

LINEAR REGRESSION OF CLASS 4 PLASTIC (36 MOVING AVERAGE).




```
Linear Regressor performance:
Mean absolute error =  26.2
Mean squared error =  1478.98
Median absolute error =  15.86
Explained variance score =  0.38
R2 score =  0.32
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

CALCULATE THE TOTAL NUMBER OF PLASTIC IN THE OCEAN BY 2050.

- By utilizing the 3rd model.

```
########## [ DEFINE MODEL AND TRAIN  < Degree = 1 > ] ##########
polynomial = PolynomialFeatures(degree=1)

X_train_transformed_for_poly = polynomial.fit_transform(X_train)

poly_linear_model = linear_model.LinearRegression()

poly_linear_model.fit(X_train_transformed_for_poly, y_train)

########## [ CALCULATE HOW MANY DAYS AFTER THE STARTING DATE ] ##########
predict_total_number_of_plastic_when = pd.to_datetime('2050/01/01')
how_many_days = predict_total_number_of_plastic_when - startdate
how_many_days = np.array(how_many_days)
print (how_many_days)
```

```
15305 days 00:00:00
```

```
how_many_days = 15305
how_many_days = np.array(how_many_days)
how_many_days = how_many_days.reshape(-1, 1)

########## [ PREDICT < Degree = 1 > ] ##########
X_test_transformed_for_poly = polynomial.fit_transform(how_many_days)

poly_y_predict = poly_linear_model.predict(X_test_transformed_for_poly)

########## [ CALCULATE THE TOTAL NUMBER OF PLASTIC BY 2050 IN THE OCEAN ] ##########
Total_Area_Ocean = 361000000

total_number_of_plastic_litter = (poly_y_predict * Total_Area_Ocean)

print ("The total area of the ocean is ", Total_Area_Ocean, ".")
print ("The count density of class 4 plastic litter predicted by 2050 = ", poly_y_predict, ".")
print ("The total number of class 4 plastic litter will be = ", total_number_of_plastic_litter, " by 2050.")
```

```
The total area of the ocean is  361000000 .
The count density of class 4 plastic litter predicted by 2050 =  [1054.80426236] .
The total number of class 4 plastic litter will be =  [3.80784339e+11]  by 2050.
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

CALCULATE THE TOTAL NUMBER OF PLASTIC IN THE OCEAN BY 2050.

- By utilizing the 4th model.

```
########## [ DEFINE MODEL AND TRAIN  < Degree = 1 > ] ##########
polynomial = PolynomialFeatures(degree=1)

X_train_transformed_for_poly = polynomial.fit_transform(X_train)

poly_linear_model = linear_model.LinearRegression()

poly_linear_model.fit(X_train_transformed_for_poly, y_train)

########## [ CALCULATE HOW MANY DAYS AFTER THE STARTING DATE ] ##########
predict_total_number_of_plastic_when = pd.to_datetime('2050/01/01')
how_many_days = predict_total_number_of_plastic_when - startdate
print (how_many_days)
```

```
15305 days 00:00:00
```

```
how_many_days = 15305
how_many_days = np.array(how_many_days)
how_many_days = how_many_days.reshape(-1, 1)

########## [ PREDICT < Degree = 1 > ] ##########
X_test_transformed_for_poly = polynomial.fit_transform(how_many_days)

poly_y_predict = poly_linear_model.predict(X_test_transformed_for_poly)

########## [ CALCULATE THE TOTAL NUMBER OF PLASTIC BY 2050 IN THE OCEAN ] ##########
Total_Area_Ocean = 361000000

total_number_of_plastic_litter = (poly_y_predict * Total_Area_Ocean)

print ("The total area of the ocean is ", Total_Area_Ocean, ".")
print ("The count density of class 4 plastic litter predicted by 2050 = ", poly_y_predict, ".")
print ("The total number of class 4 plastic litter will be = ", total_number_of_plastic_litter, " by 2050.")
```

```
The total area of the ocean is  361000000 .
The count density of class 4 plastic litter predicted by 2050 =  [945.49332808] .
The total number of class 4 plastic litter will be =  [3.41323091e+11]  by 2050.
```

REGRESSION ANALYSIS ON MARINE PLASTIC POLLUTION

IS THE PREDICTION ACCURATE ?

- Better R2 score.
- Some improvements can be observed.
- Not a perfect prediction (Suggestion of LSTM).

LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

PREDICTION WITH LSTM

- Unclear correlation.
- Sequential dataset.

```
import math
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM
import matplotlib.pyplot as plt
```

LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

PREDICTION WITH LSTM

- Scale the data.
  - MinMaxScaler.

```
scaler = MinMaxScaler(feature_range= (0,1))

scaled_data_1 = scaler.fit_transform(Dataset_1)
scaled_data_2 = scaler.fit_transform(Dataset_2)
scaled_data_3 = scaler.fit_transform(Dataset_3)
scaled_data_4 = scaler.fit_transform(Dataset_4)
```

- Split dataset.
- Train LSTM.

```
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape = (X_train_3.shape[1], 1)))
model.add(LSTM(50, return_sequences=False))
model.add(Dense(25))
model.add(Dense(1))

model.compile(optimizer = 'adam', loss = 'mean_squared_error')

model.fit(X_train_3, y_train_3, batch_size = 3, epochs = 3)
```

LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

PREDICTION WITH LSTM

- Predictions of testing dataset.
- Calculate Root Mean Square Error.

```
Predictions = model.predict(X_test_3)
Predictions = scaler.inverse_transform(Predictions)

RMSE = np.sqrt(np.mean(Predictions - y_test_3)**2)

RMSE
```

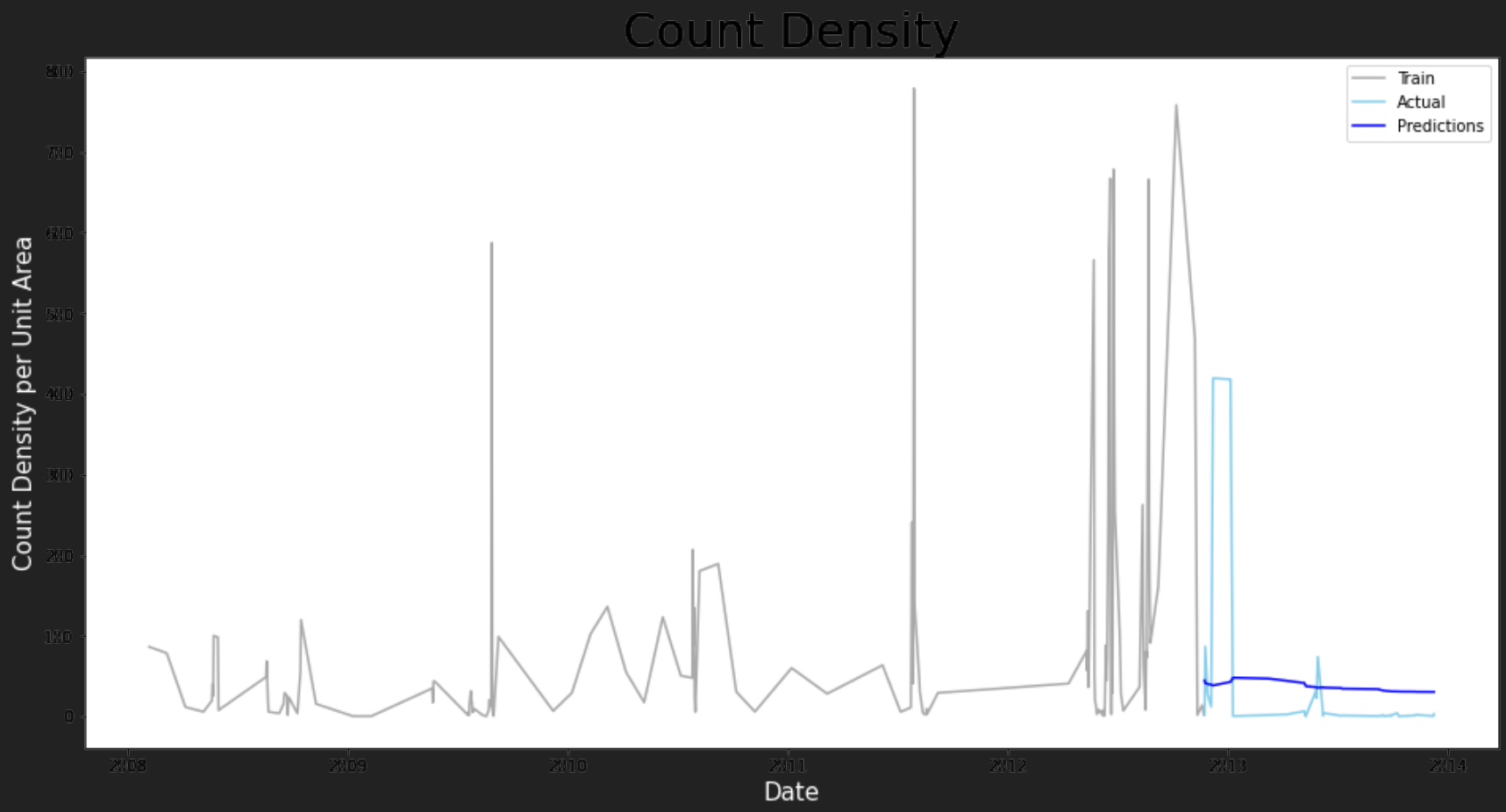
LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

PREDICTION WITH LSTM

- Root Mean Square Error
  - 5.588665…
  - 3.348325…
  - 13.97259…
  - 50.41848… respectively.

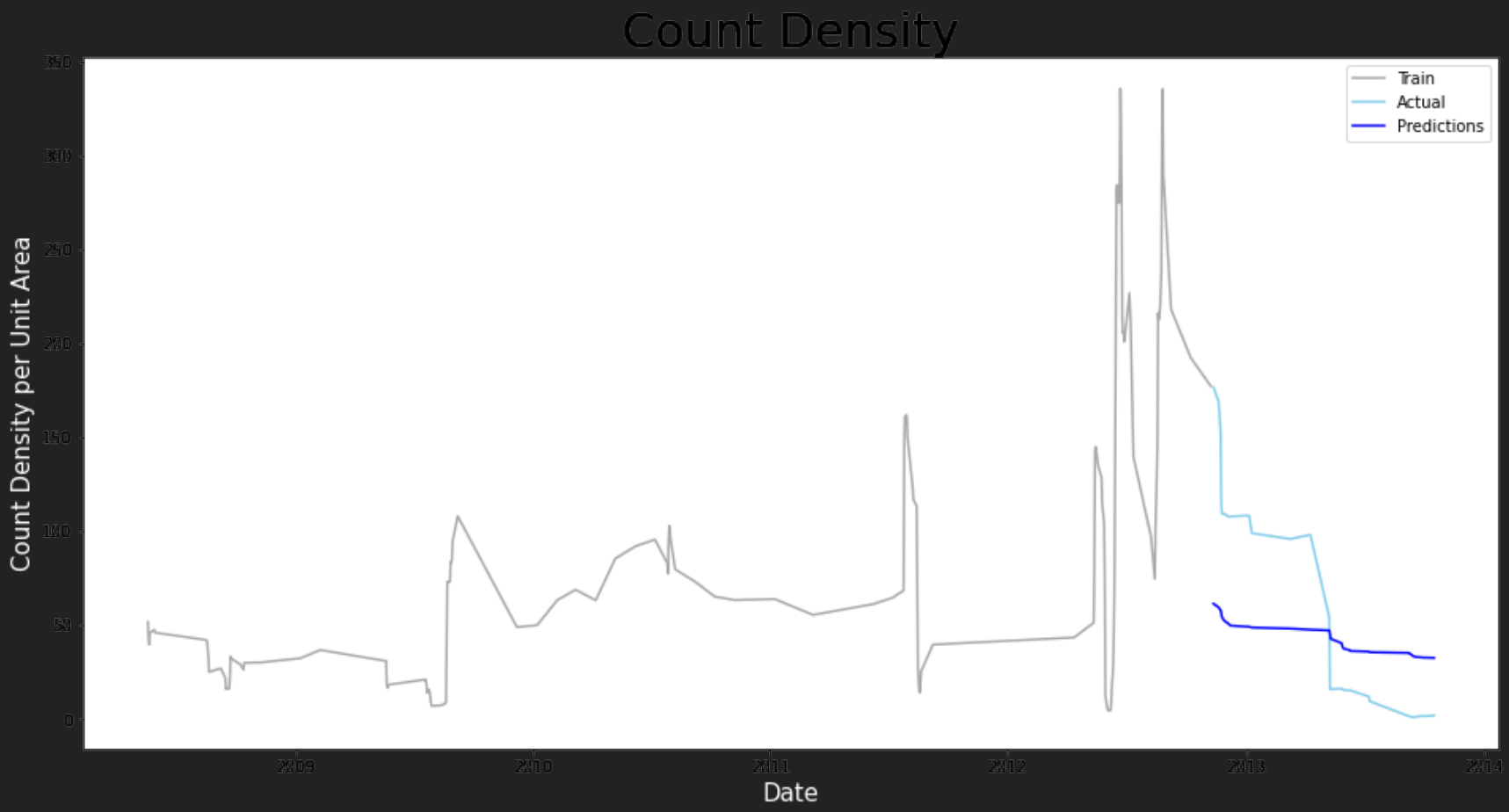
LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

LSTM PREDICTION OF RAW DATA



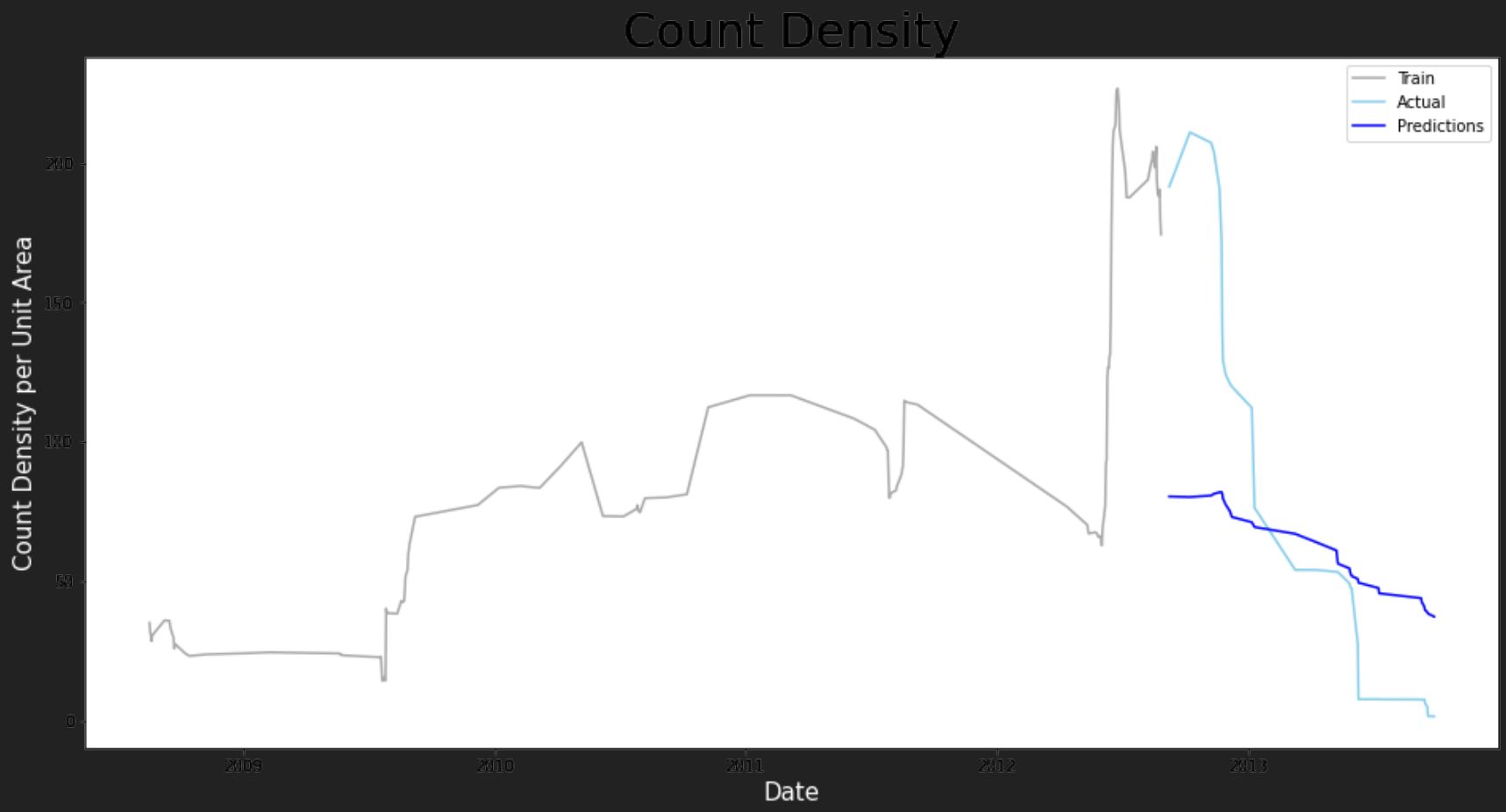
LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

LSTM PREDICTION OF MOVING AVERAGE 9



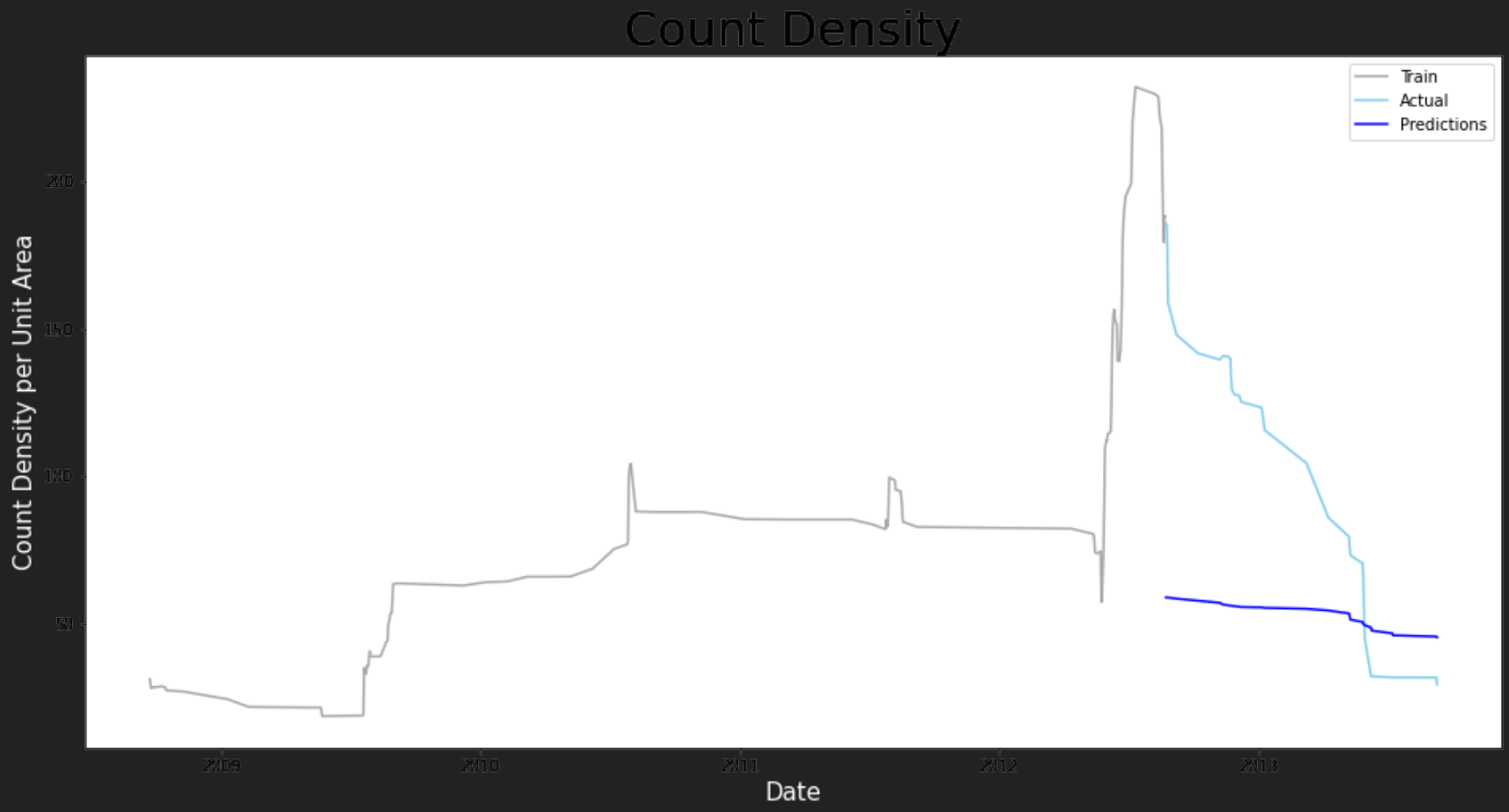
LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

LSTM PREDICTION OF MOVING AVERAGE 21



LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

LSTM PREDICTION OF MOVING AVERAGE 36



LSTM ANALYSIS ON MARINE PLASTIC POLLUTION

THE BEST PERFORMANCE OF LSTM

- By looking at RMSE values and graphs…
  - 9 moving average.
  - 21 moving average.
- LSTM prediction on the raw data and 36 moving average is not accurate.
  - Even tough, RMSE value is small in the first trained LSTM.
  - RMSE value is relatively large for the 36 moving average LSTM.

CONCLUSION ON MARINE PLASTIC POLLUTION

CONCLUSION

- Further improvement.
  - Stronger control of outliers required.
  - Stronger control of overfitting is required.
- Organization of data is important.

My code is available on :

https://drive.google.com/open?id=1iLIrnlBluqbPY_LVfBVbz8crKe9TF-R1

SAVE TURTLES

SAVE THE PLANET 

THANK YOU