Projekt 8. Przestępstwa w Chicago po raz drugi.

Nikola Girszewska

Pierwszym krokiem jest załadowanie wszystkich potrzebnych bibliotek i modułów:

```
import pandas as pd
from datetime import datetime
%matplotlib notebook
import matplotlib.pyplot as plt
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
from sklearn.linear_model import LinearRegression
```

Następnie ładuje dane potrzebne do wykonywania zadań:

In [3]:
```
dane=pd.read_csv('Crimes_-_2001_to_present.csv', sep=',')
dane.head(5)
```

Out[3]:

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Longitude	Location	Historical Wards 2003-2015
0	11677548	JC251390	05/05/2019 11:55:00 PM	013XX N DAMEN AVE	041A	BATTERY	AGGRAVATED: HANDGUN	STREET	False	False	...	-87.677323	(41.905475526, -87.677323168)	24.0
1	11677416	JC251398	05/05/2019 11:53:00 PM	050XX W VAN BUREN ST	143A	WEAPONS VIOLATION	UNLAWFUL POSS OF HANDGUN	STREET	True	False	...	-87.750290	(41.874908618, -87.750289695)	52.0
2	11677463	JC251408	05/05/2019 11:46:00 PM	024XX E 77TH ST	502P	OTHER OFFENSE	FALSE/STOLEN /ALTERED TRP	STREET	True	False	...	-87.564785	(41.755379832, -87.564785446)	43.0
3	11677501	JC251378	05/05/2019 11:44:00 PM	057XX S WOLCOTT AVE	041A	BATTERY	AGGRAVATED: HANDGUN	SIDEWALK	False	False	...	-87.671746	(41.789399707, -87.671746191)	44.0
4	11677483	JC251406	05/05/2019 11:40:00 PM	092XX S KINGSTON AVE	0498	BATTERY	AGGRAVATED DOMESTIC BATTERY: HANDS/FIST/FEET S...	RESIDENCE	False	True	...	-87.562088	(41.727526285, -87.562087901)	43.0

5 rows × 30 columns

Zadanie 1.

Oblicz, ile przestępstw popełniono w każdym miesiącu każdego roku (od 2001 do 2019). Narysuj wykres liniowy prezentujący te liczby.

In [4]:
```
#ZADANIE 1
daty = dane.Date
#print(daty)
```

In [6]:
```
dat1 = [datetime.strptime(daty[i][:10], '%m/%d/%Y') for i in range(len(daty))]
dat2 = [(dat1[i].year, dat1[i].month) for i in range(len(dat1))]
tabela = pd.value_counts(dat2) #tworzymy zbiór danych zawierający liczbe przestępstw w każdym roku każdego miesiąca
print(tabela)
```

```
(2002, 7)     46013
(2001, 7)     44692
(2003, 8)     44268
(2002, 8)     44210
(2001, 8)     44032
(2003, 7)     43415
(2003, 10)    43327
(2004, 7)     43236
(2002, 10)    43145
(2004, 8)     43044
(2001, 10)    43029
(2002, 5)     42913
(2002, 6)     42834
(2002, 9)     42388
(2001, 5)     41821
(2005, 7)     41806
(2001, 6)     41725
(2006, 7)     41547
(2005, 8)     41543
```

In [7]:
```
#z powyższych danych tworzymy ramke danych
RM_t = tabela.index.tolist()
LP_t = []
for i in range(len(tabela)):
    LP_t.append(tabela[i])

ramka_t = pd.DataFrame([RM_t,LP_t],index=['RM','LP'])
ramka_n = ramka_t.T
print(ramka_n)
```

```
              RM     LP
0     (2002, 7)  46013
1     (2001, 7)  44692
2     (2003, 8)  44268
3     (2002, 8)  44210
4     (2001, 8)  44032
5     (2003, 7)  43415
6    (2003, 10)  43327
7     (2004, 7)  43236
8    (2002, 10)  43145
9     (2004, 8)  43044
10   (2001, 10)  43029
11    (2002, 5)  42913
12    (2002, 6)  42834
13    (2002, 9)  42388
14    (2001, 5)  41821
15    (2005, 7)  41806
16    (2001, 6)  41725
17    (2006, 7)  41547
```

In [8]:
```
#tworzymy posortowaną według dat ramke danych
one = ramka_n['RM'].sort_values()
two = ramka_n['LP']
R = pd.concat([one,two],axis=1)
ramka_S=R.sort_values(by=['RM'])
print(ramka_S)
```

```
                RM     LP
54     (2001, 1)  38100
91     (2001, 2)  33779
28     (2001, 3)  40552
34     (2001, 4)  40080
14     (2001, 5)  41821
16     (2001, 6)  41725
1      (2001, 7)  44692
4      (2001, 8)  44032
20     (2001, 9)  41502
10    (2001, 10)  43029
43    (2001, 11)  39596
65    (2001, 12)  36846
51     (2002, 1)  38401
88     (2002, 2)  33909
50     (2002, 3)  38584
36     (2002, 4)  40035
11     (2002, 5)  42913
12     (2002, 6)  42834
```

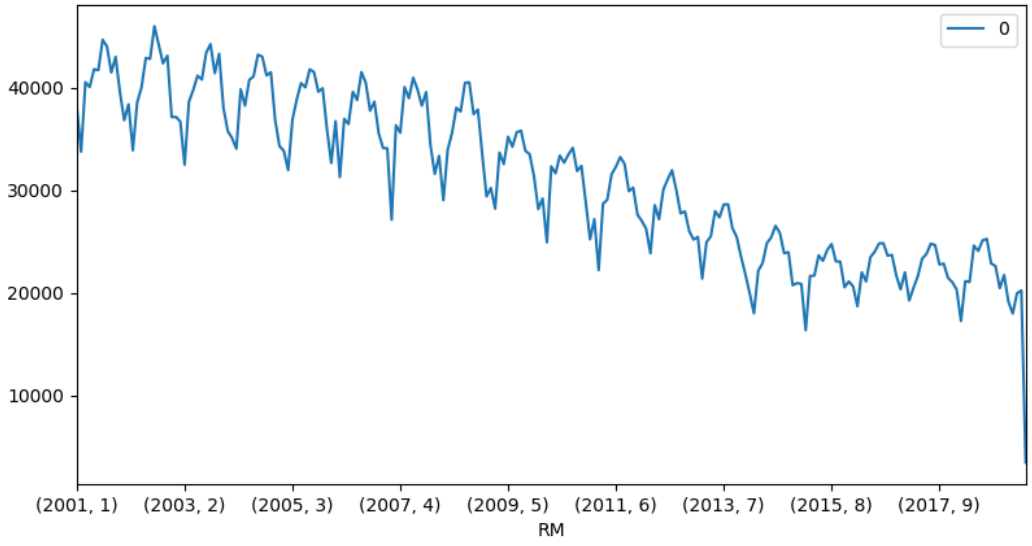
In [13]:
```
#tworzymy ramke tanych tak, aby data była indeksem
y_lab=ramka_S['LP'].tolist()
ramka_nowa=pd.DataFrame(y_lab,index=ramka_S['RM'])
ramka_nowa
```

Out[13]:

RM	0
(2001, 1)	38100
(2001, 2)	33779
(2001, 3)	40552
(2001, 4)	40080
(2001, 5)	41821
(2001, 6)	41725
(2001, 7)	44692
(2001, 8)	44032
(2001, 9)	41502
(2001, 10)	43029

Tak przygotowane dane można przedstawić na wykresie.

In [14]:
```
#rysujemy wykres przedstawiający liczbę przestępstw w każdym miesiącu każdego roku od 2001 do 2019
ramka_nowa.plot()
```



Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x260fe648908>

Wykres ten przedstawia, sposób w jaki zmieniały się liczby przestępstw w każdym miesiącu od 2001 do 2019 roku. Możemy zauważyć na nim znaczny spadek liczby przestępstw.

By móc dokładnie odczytywać ile przestępstw popełniono w każdym miesiącu każdego roku, stworzyłam ramkę danych zawierającą te dane. Wygląda ona tak:

	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
STYCZEŃ	38100	38401	36713	35105	33861	36752	34100	33366	30236	29210	27192	26265	25461	19994	20867	20630	22018	20337	19143
LUTY	33779	33909	32509	34066	31985	31298	27138	29044	28227	24942	22228	23869	21394	18014	16359	18694	19264	17263	17995
MARZEC	40552	38584	38650	39855	36906	36983	36348	33977	33683	32335	28694	28562	24958	22175	21645	22011	20509	21157	19975
KWIECIEŃ	40080	40035	39789	38268	38870	36475	35631	35592	32566	31668	29101	27178	25521	22900	21689	21118	21646	21061	20238
MAJ	41821	42913	41191	40779	40467	39613	40085	38063	35238	33397	31583	30088	27981	24853	23663	23490	23336	24623	3455
CZERWIEC	41725	42834	40815	41134	40050	38812	38993	37698	34266	32712	32310	31075	27362	25416	23136	24016	23800	24094	0
LIPIEC	44692	46013	43415	43236	41806	41547	40987	40491	35675	33510	33253	31969	28621	26538	24192	24824	24805	25122	0
SIERPIEŃ	44032	44210	44268	43044	41543	40496	39848	40538	35826	34147	32582	30034	28640	25863	24769	24856	24666	25273	0
WRZESIEŃ	41502	42388	41423	41213	39623	37771	38264	37444	33869	31887	29927	27750	26327	23874	23074	23639	22777	22863	0
PAŹDZIERNIK	43029	43145	43327	41523	39961	38659	39590	37884	33530	32390	30270	27958	25459	23979	23052	23705	22858	22631	0
LISTOPAD	39596	37152	38055	36838	35977	35567	34422	33555	31388	28883	27632	26025	23551	20743	20552	21711	21462	20468	0
GRUDZIEŃ	36846	37172	35791	34322	32668	34140	31609	29412	28185	25237	27018	25207	21849	20964	21118	20347	21040	21748	0

Zadanie 2.
Spośród przestępstw, w których doszło do aresztowania (Arrest), wybierz najczęściej popełniane w każdym roku.

```
In [15]: # ZADANIE 2
aresztowania=dane[dane['Arrest']==True] #wybieram do dalszej analizy przestępstwa, w których doszło do aresztowania
aresztowania.head(10)
```

21	11677437	JC251381	05/05/2019 11:13:00 PM	023XX S DRAKE AVE	143A	WEAPONS VIOLATION	UNLAWFUL POSS OF HANDGUN	STREET	True	False	...	-87.713698	(41.848888074, -87.713698143)	14.
22	11677430	JC251365	05/05/2019 11:12:00 PM	026XX W 51ST ST	143A	WEAPONS VIOLATION	UNLAWFUL POSS OF HANDGUN	STREET	True	False	...	-87.689660	(41.801063154, -87.689659697)	49.
25	11677366	JC251357	05/05/2019 11:05:00 PM	012XX S AVERS AVE	2027	NARCOTICS	POSS: CRACK	STREET	True	False	...	-87.721502	(41.865763196, -87.721501887)	36.
36	11677445	JC251395	05/05/2019 11:00:00 PM	077XX S ESSEX AVE	0486	BATTERY	DOMESTIC BATTERY SIMPLE	APARTMENT	True	False	...	-87.563865	(41.754808347, -87.563865036)	43.
37	11677422	JC251362	05/05/2019 10:58:00 PM	007XX W 66TH PL	143A	WEAPONS VIOLATION	UNLAWFUL POSS OF HANDGUN	SIDEWALK	True	False	...	-87.643440	(41.773406885, -87.643440433)	31.

10 rows × 30 columns

In [16]:
```
for i in range(0,19): #tworzę pętle
    areszt=aresztowania[aresztowania['Year']==2001+i]
    a=pd.value_counts(areszt['Description']) #licze wystąpienia poszczególnych przestępstw w danym roku (od najczęściej popeł
    b=areszt['Description'].value_counts().keys() #wyodrębniam nazwy przestępstw w danym roku (od najczęściej popełnianych)
    print("Najczęściej popełnianym przestępstwem w roku", 2001+i, "było:",b[0] ,"w liczbie", a[0])
```

```
Najczęściej popełnianym przestępstwem w roku 2001 było: SIMPLE w liczbie 20614
Najczęściej popełnianym przestępstwem w roku 2002 było: SIMPLE w liczbie 16535
Najczęściej popełnianym przestępstwem w roku 2003 było: POSS: CANNABIS 30GMS OR LESS w liczbie 17556
Najczęściej popełnianym przestępstwem w roku 2004 było: POSS: CANNABIS 30GMS OR LESS w liczbie 18780
Najczęściej popełnianym przestępstwem w roku 2005 było: POSS: CANNABIS 30GMS OR LESS w liczbie 19210
Najczęściej popełnianym przestępstwem w roku 2006 było: POSS: CANNABIS 30GMS OR LESS w liczbie 20403
Najczęściej popełnianym przestępstwem w roku 2007 było: POSS: CANNABIS 30GMS OR LESS w liczbie 22819
Najczęściej popełnianym przestępstwem w roku 2008 było: POSS: CANNABIS 30GMS OR LESS w liczbie 20405
Najczęściej popełnianym przestępstwem w roku 2009 było: POSS: CANNABIS 30GMS OR LESS w liczbie 21177
Najczęściej popełnianym przestępstwem w roku 2010 było: POSS: CANNABIS 30GMS OR LESS w liczbie 21979
Najczęściej popełnianym przestępstwem w roku 2011 było: POSS: CANNABIS 30GMS OR LESS w liczbie 20103
Najczęściej popełnianym przestępstwem w roku 2012 było: POSS: CANNABIS 30GMS OR LESS w liczbie 17689
Najczęściej popełnianym przestępstwem w roku 2013 było: POSS: CANNABIS 30GMS OR LESS w liczbie 15944
Najczęściej popełnianym przestępstwem w roku 2014 było: POSS: CANNABIS 30GMS OR LESS w liczbie 12876
Najczęściej popełnianym przestępstwem w roku 2015 było: POSS: CANNABIS 30GMS OR LESS w liczbie 9896
Najczęściej popełnianym przestępstwem w roku 2016 było: DOMESTIC BATTERY SIMPLE w liczbie 5126
Najczęściej popełnianym przestępstwem w roku 2017 było: RETAIL THEFT w liczbie 4920
Najczęściej popełnianym przestępstwem w roku 2018 było: DOMESTIC BATTERY SIMPLE w liczbie 5088
Najczęściej popełnianym przestępstwem w roku 2019 było: DOMESTIC BATTERY SIMPLE w liczbie 1660
```

Zadanie 3.

Zaimplementuj samodzielnie funkcje wyznaczającą współczynniki prostej do regresji liniowej. Z wykorzystaniem tej funkcji zbuduj model regresji liniowej przewidujący liczbę przestępstw w kolejnych latach. Dokładniej - zbuduj kilka modeli (dokonując różnego podziału na zbiór uczący i testowy) i wybierz najkorzystniejszy. Na tej podstawie oszacuj liczbę aresztowań w 2019 roku. Ponadto, na podstawie danych z pliku oraz przewidywań, oblicz ile średnio aresztowań zostanie przeprowadzonych w każdym pozostałym miesiącu 2019 roku.

Poniższy model przedstawia liczbę oszacowaną liczbę przestępstw dla 2020 roku, jednak program działa tak, że możemy podać dowolny rok.

In [17]:
```
# ZADANIE 3

#piszsemy funkcję wyznaczającą współczynniki regresji liniowej
def regresja(X,Y):
    suma1=0
    suma2=0
    sr_X=np.mean(X)
    sr_Y=np.mean(Y)
    for i in range(0,len(X)):
        S1=(X[i]-sr_X)*(Y[i]-sr_Y)
        suma1=suma1+S1
    for j in range(0,len(X)):
        S2=(X[j]-sr_X)**2
        suma2=suma2+S2
    a=suma1/suma2
    b=sr_Y-(a*sr_X)
    return(a,b)
```

In [18]:
```
nowe_dane=pd.DataFrame(dane['Year'],index=dane.index)
```

In [19]:
```
X=[] #tworzymy liste która zawierać będzie roki
for i in range(2001,2019):
    X.append(i)
print(X)

Y=[] #tworzymy liste, która zawiera ilości popełnionych przestępstw od roku 2001 do 2018
for i in range(2001,2019):
    N=nowe_dane[nowe_dane['Year']==i]
    Y.append(len(N))
print(Y)
```

```
[2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018]
[485754, 486756, 475946, 469383, 453717, 448113, 437015, 427064, 392689, 370318, 351790, 335980, 307124, 275313, 264116,
269041, 268181, 266640]
```

In [22]:
```
print(regresja(X,Y))
```

```
(-15786.575851393189, 32100065.284485724)
```

In [79]:
```
def R2(x, y, deg=1):
    wynik = {}
    C = np.corrcoef(x, y)[0,1]
    wynik['R2'] = C**2
    return wynik['R2']
```

In [171]:
```
proby = []
n= int(input("Podaj liczbę testów: "))
for i in range(0,n):
    X_uczacy, X_testowy, Y_uczacy, Y_testowy = train_test_split(X, Y, test_size = 0.2 )
    reg = regresja(X_uczacy,Y_uczacy)
    R_2 = R2(X_testowy,Y_testowy)
    Z=[R_2,reg]
    proby.append(Z)
M=(max(proby))
print("Najkorzystniejszy model ma współczynnik determinacji i współczynniki regresji liniowej równe: ",M)
rok = int(input("Podaj rok, w którym chcesz przewidzieć liczbę przestępstw: "))
print("Na podstawie tych danych w ",rok, "roku dojdzie do", round(M[1][0]*rok+M[1][1]), "przestępstw.")
```

```
Podaj liczbę testów: 10
Najkorzystniejszy model ma współczynnik determinacji i współczynniki regresji liniowej równe:  [0.9965672343817935,
(-16651.1520979021, 33833386.59265735)]
Podaj rok, w którym chcesz przewidzieć liczbę przestępstw: 2020
Na podstawie tych danych w  2020 roku dojdzie do 198059.0 przestępstw.
```

Poniższy kod szacuje liczbę aresztowań, do których dojdzie w 2019 roku.

In [231]:
```
#oszacujemy liczbę aresztowań w 2019
ar_liczba=pd.value_counts(aresztowania['Year'])
```

In [238]:
```
X_a = ar_liczba.index.tolist()
del X_a[18]
Y_a = ar_liczba.tolist()
del Y_a[18]
```

In [239]:
```
proby_ar = []
n_a= int(input("Podaj liczbę testów: "))
for i in range(0,n_a):
    X_uczacy, X_testowy, Y_uczacy, Y_testowy = train_test_split(X_a, Y_a, test_size = 0.2 )
    reg = regresja(X_uczacy,Y_uczacy)
    R_2 = R2(X_testowy,Y_testowy)
    Z=[R_2,reg]
    proby_ar.append(Z)
M_a=(max(proby_ar))
print("Na podstawie danych w 2019 roku dojdzie do", round(M_a[1][0]*2019+M_a[1][1]), "arezstowań.")
```

```
Podaj liczbę testów: 10
Na podstawie danych w 2019 roku dojdzie do 47660.0 arezstowań.
```

Podjęłam również próbę przewidzenia ile średnio aresztowań zostanie przeprowadzonych w każdym pozostałym miesiącu 2019 roku.

In [34]:
```
l_arest = aresztowania[aresztowania['Year']==2019]
l_arest_2019 = pd.DataFrame(l_arest['Date'])
l_arest_2019['month']=pd.DatetimeIndex(l_arest_2019['Date']).month
l_arest_2019.head(5)
```

Out[34]:

	Date	month
1	05/05/2019 11:53:00 PM	5
2	05/05/2019 11:46:00 PM	5
8	05/05/2019 11:33:00 PM	5
15	05/05/2019 11:29:00 PM	5
18	05/05/2019 11:20:00 PM	5

In [35]:
```
T = pd.value_counts(l_arest_2019['month'])
T
```

Out[35]:
```
3    4479
1    4225
4    4111
2    3931
5     648
Name: month, dtype: int64
```

In [36]:
```
T_m = T.index.tolist()
print(T_m)
T_l = T.tolist()
print(T_l)
```

```
[3, 1, 4, 2, 5]
[4479, 4225, 4111, 3931, 648]
```

Niestety model ten nie działa poprawnie.

In [41]:
```
proby_T = []
n= int(input("Podaj liczbę testów: "))
for i in range(0,n):
    X_uczacy, X_testowy, Y_uczacy, Y_testowy = train_test_split(T_m, T_l, test_size = 0.2 )
    reg = regresja(X_uczacy,Y_uczacy)
    R_2 = R2(X_testowy,Y_testowy)
    Z=[R_2,reg]
    proby.append(Z)
M_T=(max(proby))
print(M_T,M_T[1][0],M_T[1][1])

for i in range (6,13):
    print("Na podstawie tych danych w ",i, "miesiącu roku 2019 dojdzie do", round(M_T[1][0]*i+M_T[1][1]), "aresztowań.")
```

```
Podaj liczbę testów: 14
[0.9891305531322341, (-16563.83838150289, 33660724.248554915)] -16563.83838150289 33660724.248554915
Na podstawie tych danych w  6 miesiącu roku 2019 dojdzie do 33561341.0 aresztowań.
Na podstawie tych danych w  7 miesiącu roku 2019 dojdzie do 33544777.0 aresztowań.
Na podstawie tych danych w  8 miesiącu roku 2019 dojdzie do 33528214.0 aresztowań.
Na podstawie tych danych w  9 miesiącu roku 2019 dojdzie do 33511650.0 aresztowań.
Na podstawie tych danych w  10 miesiącu roku 2019 dojdzie do 33495086.0 aresztowań.
Na podstawie tych danych w  11 miesiącu roku 2019 dojdzie do 33478522.0 aresztowań.
Na podstawie tych danych w  12 miesiącu roku 2019 dojdzie do 33461958.0 aresztowań.
```

Zadanie 4.
Wykonaj Zadanie 3 wykorzystując bibliotekę Sklearn i wbudowana w niej możliwość wykorzystania regresji liniowej. Porównaj wyniki.

In [219]:
```
# ZADANIE 4

X1 = np.array([[2001], [2002], [2003], [2004], [2005], [2006], [2007], [2008], [2009], [2010], [2011], [2012], [2013], [2014]
Y1 = np.array([485754, 486756, 475946, 469383, 453717, 448113, 437015, 427064, 392689, 370318, 351790, 335980, 307124, 275313

S=[]
n1= int(input("Podaj liczbę testów: "))
for i in range(0,n1):
    X_uczacy, X_testowy, Y_uczacy, Y_testowy = train_test_split(X1, Y1, test_size = 0.2 )
    model = LinearRegression()
    model.fit(X_uczacy,Y_uczacy)
    Y_przewidywane = model.predict(X_testowy)
    ER2 = r2_score(Y_testowy, Y_przewidywane)
    x = model.coef_
    y = model.intercept_
    W=[ER2,x,y]
    S.append(W)
M_s = max(S)

print("Najkorzystniejszy model ma współczynnik determinacji i współczynniki regresji liniowej równe: ",M_s)
rok = int(input("Podaj rok, w którym chcesz przewidzieć liczbę przestępstw: "))
print("Na podstawie tych danych w ",rok, "roku dojdzie do", M_s[1]*rok+M_s[2], "przestępstw.")
```

```
Najkorzystniejszy model ma współczynnik determinacji i współczynniki regresji liniowej równe:  [0.9866342013689939, array
([-16017.25343393]), 32564226.20700328]
Podaj rok, w którym chcesz przewidzieć liczbę przestępstw: 2019
Na podstawie tych danych w  2019 roku dojdzie do [225391.52389244] przestępstw.
```

In [3]:

```
X1_a = np.array([[2004], [2001], [2003], [2002], [2005], [2006], [2007], [2009], [2008], [2010], [2011], [2012], [2013], [201
Y1_a = np.array([144686, 141903, 141572, 141555, 140895, 135387, 131852, 110769, 109961, 100484, 96230, 90575, 86462, 79539,

J = []
n2= int(input("Podaj liczbę testów: "))
for i in range(0,n2):
    X_uczacy, X_testowy, Y_uczacy, Y_testowy = train_test_split(X1_a, Y1_a, test_size = 0.2 )
    model = LinearRegression()
    model.fit(X_uczacy,Y_uczacy)
    Y_przewidywane = model.predict(X_testowy)
    ER2 = r2_score(Y_testowy, Y_przewidywane)
    x = model.coef_
    y = model.intercept_
    L=[ER2,x,y]
    J.append(L)
M_j = max(J)
print("Na podstawie danych w 2019 roku dojdzie do", M_j[1]*2019+M_j[2], "arezstowań.")
```

```
Podaj liczbę testów: 10
Na podstawie danych w 2019 roku dojdzie do [46915.51302254] arezstowań.
```

Wyniki tego szacunku oraz oszacowania z zadania 3. są do siebie bardzo podobne, co wskazuje na to, że oba sposoby przewidywania są poprawne i ciężko wybrać lepszy.