Wykład Specjalizujący

Zadanie 2
Regresja liniowa

Autor:

Jerzy Dębowski 151266

Plik z danymi

Do zadania wykorzystałem plik zawierający dane odnośnie średniej temperatury w styczniu w latach 1953-2018 w Nowym Jorku, który można pobrać ze strony:
https://www.ncdc.noaa.gov/cag/

Załadowanie pliku

```
10    temperatures_df = pd.read_csv( filepath_or_buffer: 'data.csv', sep=',', header=4)
```

Przygotowanie danych

Nazwy kolumn zostały zmienione na: “Data”, “Temperatura” oraz “Odchyłka”.

- W kolumnie “Data” wartości zostały zamienione na sam rok przy użyciu funkcji to_datetime z biblioteki pandas
- W kolumnach “Temperatura” oraz “Odchyłka” dane w stopniach Fahrenheit’a zostały zmienione na wartości w stopniach Celsjusza przy użyciu funkcji fahrenheit_to_celsius w zaokrągleniu do 2. miejsca po przecinku

```
    3 usages   new *
6   def fahrenheit_to_celsius(temperature):
7       return round((temperature - 32) * 5 / 9, 2)
8
9
10  temperatures_df = pd.read_csv( filepath_or_buffer: 'data.csv', sep=',', header=4)
11  temperatures_df.columns = [
12      'Data',
13      'Temperatura',
14      'Odchyłka',
15  ]
16  temperatures_df['Data'] = temperatures_df['Data'].apply(
17      lambda x: pd.to_datetime(str(x), format='%Y%m').year
18  )
19  temperatures_df['Temperatura'] = temperatures_df['Temperatura'].apply(
20      lambda x: fahrenheit_to_celsius(x)
21  )
22  temperatures_df['Odchyłka'] = temperatures_df['Odchyłka'].apply(
23      lambda x: fahrenheit_to_celsius(x)
24  )
25
```

Opis danych

Opis danych otrzymujemy poprzez wywołanie na ramce danych funkcji describe(). Otrzymujemy między innymi średnią, maksymalną lub minimalną wartość w każdej kolumnie:

```
25
26      temperatures_df_description = temperatures_df.describe()
27      print(f'Opis ramki danych:\n{temperatures_df_description}')
28
```

```
Opis ramki danych:
              Data  Temperatura   Odchyłka
count    66.000000    66.000000  66.000000
mean   1985.500000    -0.668636 -17.399848
std      19.196354     1.740301   1.740420
min    1953.000000    -5.610000 -22.340000
25%    1969.250000    -1.432500 -18.162500
50%    1985.500000    -0.580000 -17.310000
75%    2001.750000     0.562500 -16.167500
max    2018.000000     3.850000 -12.880000
```

Regresja liniowa

Regresję wyliczamy przy użyciu funkcji linregress z biblioteki stats. Jako x podajemy kolumnę z rokiem, a jako y kolumnę z temperaturami:

```
slope, intercept, r_value, p_value, std_err = stats.linregress(
    x=temperatures_df['Data'],
    y=temperatures_df['Temperatura']
)
```

Prognozowaną temperaturę dla stycznia 2022 roku wyliczamy predykcję dla wartości 2022:

```
probably_january_2022_temperature = round(slope * 2022 + intercept, 2)
```

Według strony, z której zostały pobrane dane, średnia temperatura w styczniu 2022 roku wynosiła 31.17 stopni Fahrenheit'a (-0.46 stopni Celsjusza)

January 2022	31.17°F

Porównanie prognozowanej temperatury z faktycznym stanem:

```
probably_january_2022_temperature = round(slope * 2022 + intercept, 2)
january_2022_temperature = fahrenheit_to_celsius(31.17)
print(
    f'Prognozowana średnia temperatura w styczniu 2022: {probably_january_2022_temperature}\n'
    f'Faktyczna średnia temperatura w styczniu 2022: {january_2022_temperature}'
)
```

```
Prognozowana średnia temperatura w styczniu 2022: 0.56
Faktyczna średnia temperatura w styczniu 2022: -0.46
```

Otrzymujemy różnicę o 1.02 stopnia Celsjusza, co potwierdza, że prognozowanie oparte na modelu liniowym obarczone jest błędem.

Wizualizacja

Do wizualizacji wykorzystano bibliotekę matplotlib. Najpierw zaznaczono punkty rzeczywistych średnich temperatur w styczniu w latach 1953-2018, a następnie predykcję dla wszystkich dat:

```
y_pred = slope * temperatures_df['Data'] + intercept
plt.scatter(temperatures_df['Data'], temperatures_df['Temperatura'], label='Dane rzeczywiste')
plt.plot( *args: temperatures_df['Data'], y_pred, color='red', label='Regresja liniowa')
plt.xlabel('Rok')
plt.ylabel('Temperatura')
plt.legend()
plt.title('Regresja liniowa w szeregu czasowym')
plt.show()
```