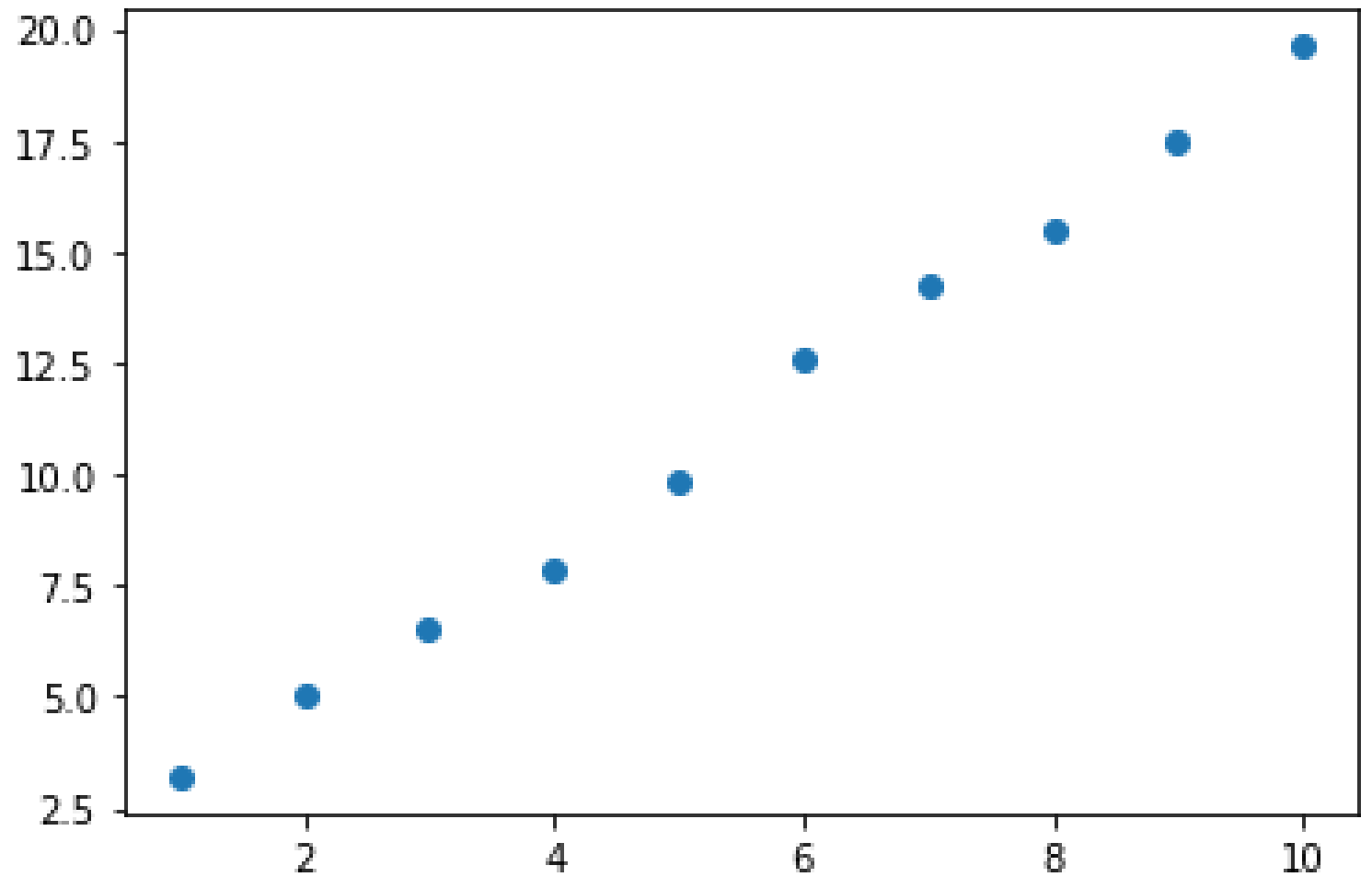


Środowiska uruchomieniowe AutoML

Lab07: Refaktoryzacja kodu

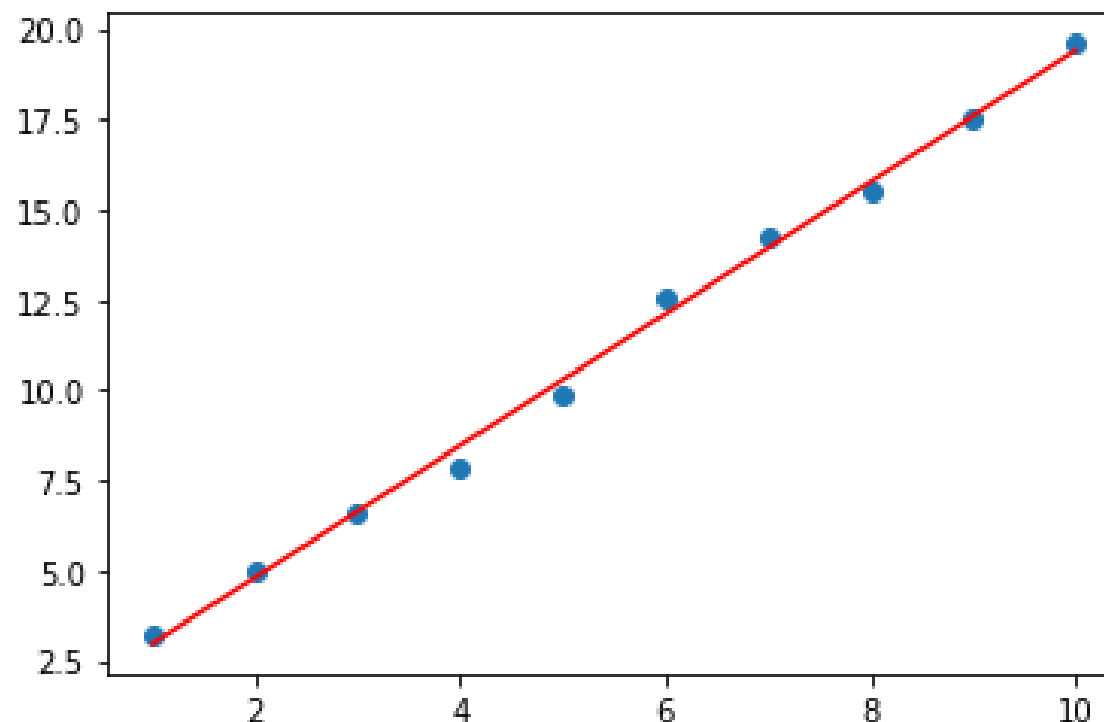
Regresja liniowa

Zbiór danych
charakteryzowanych przez
dwie wartości x i y



Regresja liniowa

Regresja liniowa jest metodą poszukującą takiej funkcji liniowej, która jak najdokładniej może przewidywać rozkład bieżących i przyszłych wartości

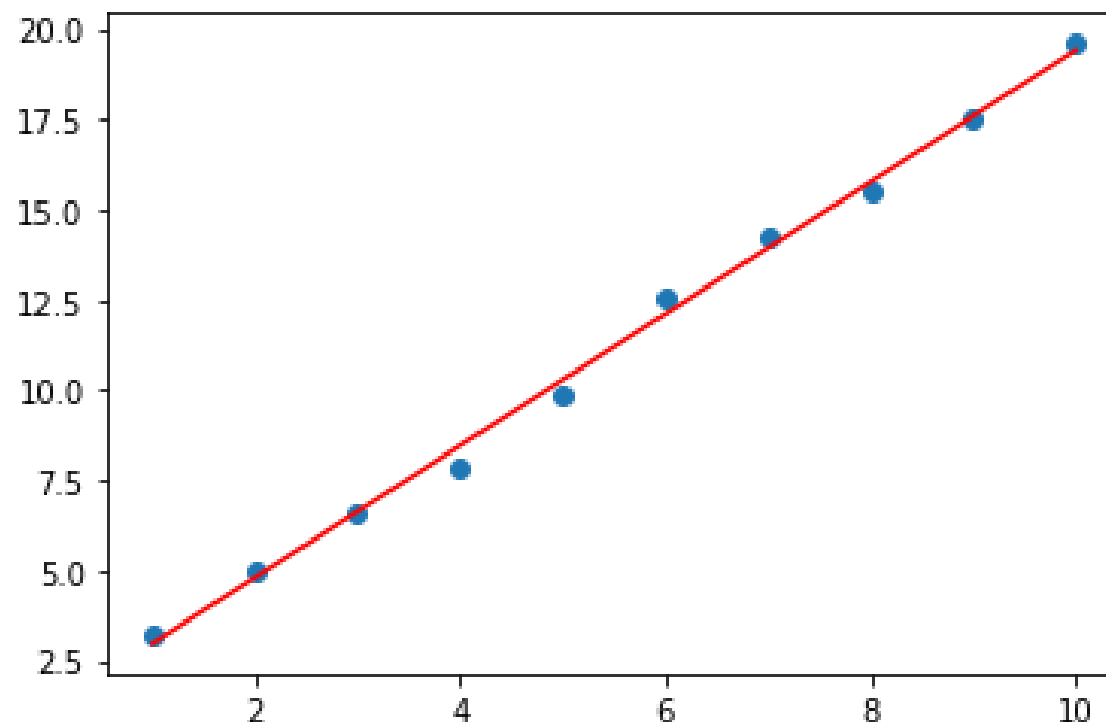


Regresja liniowa

Funkcja liniowa ma postać:

$$y = a * x + b$$

Uczenie się algorytmu w regresji liniowej polega na poszukiwaniu współczynników a i b przy pomocy różnego rodzaju metod



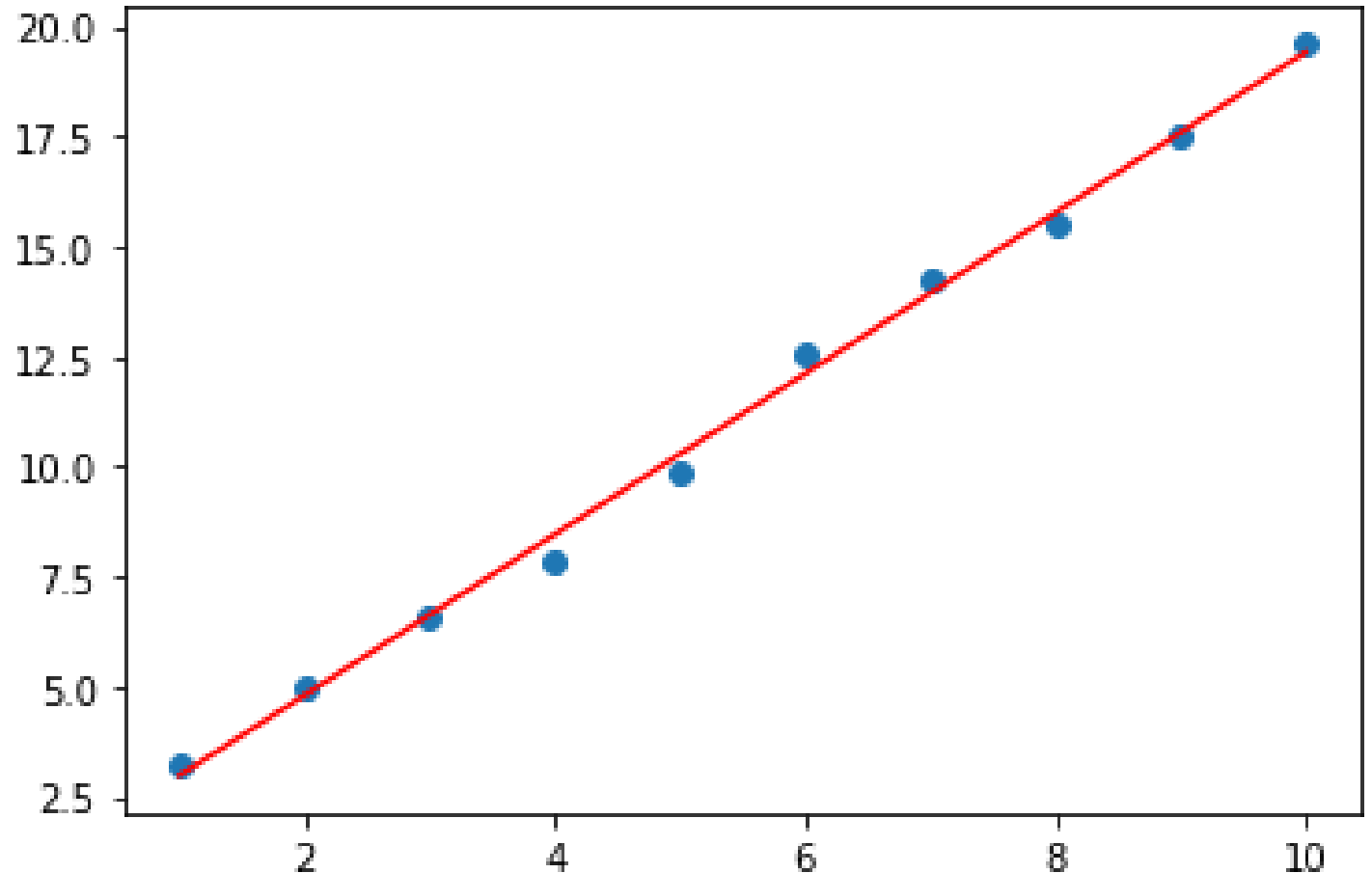
Regresja liniowa

Funkcja liniowa ma postać:

$$y = a * x + b$$

Wartość a (coef) odpowiada za nachylenie wykresu funkcji

Wartość b (intercept) odpowiada za położenie względem osi



Regresja liniowa – Jupyter Notebook

Otrzymaliśmy 2 pliki Jupyter Notebook zawierające rozważania dotyczące wyznaczenia algorytmu uczącego się na podstawie danych zawartych w pliku csv.

Regresja liniowa – Jupyter Notebook

1_10_points_training.ipynb

Autor odnalazł najskuteczniejszą metodę uczenia się - regresję liniową, następnie wykonał trening algorytmu na podstawie danych i sprawdził przewidywanie.

W celu sprawdzenia poprawności działania modelu wyodrębnił z modelu współczynniki a (coef) i b (intercept) i wykonał obliczenie funkcji dla zadanego x . Wynik pokrył się z przewidywaniem.

Na końcu wytrenowany model zapisał w pliku serializacyjnym pickle.

Regresja liniowa – Jupyter Notebook

2_10_points_inference.ipynb

Autor otworzył plik z modelem w formie pickle i ponownie wykonał proces trenowania i przewidywania dla zadanej wartości x w ramach modelu liniowego:

$$y = a * x + b$$

W celu sprawdzenia poprawności działania modelu wyodrębnił z modelu współczynniki a (coef) i b (intercept), żeby sprawdzić, czy wartości są takie same jak w przypadku pliku pierwszego.

Zadanie - wymagania

Naszym celem jest przygotowanie kodu w czystym Pythonie, który będzie można wykorzystać w procesie tworzenia aplikacji webowej.

Aplikacja będzie potrafiła:

- dokonywać przewidywania na podstawie danych otrzymanych od użytkownika
- ulepszać model na podstawie dostarczanych nowych danych

Zadanie - treść

Proszę o przygotowanie dwóch funkcji napisanych w czystym Pythonie:

1. Funkcja I przyjmuje dowolną wartość x i dzięki wykorzystaniu modelu w pliku serializacyjnym **pickle: our_model.pkl** zwraca wartość y (przewiduje).
2. Funkcja II przyjmuje dowolne wartości x i y , zapisuje je na końcu pliku csv z danymi **10_points.csv**, trenuje model nowymi danymi, nadpisuje stary model **our_model.pkl** nowym plikiem.

Inaczej mówiąc przygotowujemy funkcję, która będzie aktualizować model na podstawie nowych danych.

Rozwiązanie zadania – kroki działania

1. Zapoznajemy się z notebookami
2. Kopiujemy fragmenty, które są dla nas przydatne
3. Wyrzucamy niepotrzebne

Funkcja pierwsza

1. Otwieramy plik **pickle** i przypisujemy do zmiennej, która będzie przechowywała model
2. Dokonujemy przewidywania wartości y na podstawie dostarczonego x
3. Zwracamy y

Przydatny kod dla funkcji I

```
1 import numpy as np
2 import pandas as pd
3 import pickle
4
5 imported_model = pickle.load(open("our_model.pkl", 'rb'))
6
7 X_unknown = np.array([2.78])
8
9 y_unknown = imported_model.predict(X_unknown.reshape(-1,1))
10
11 print('The question: X = ',X_unknown[0])
12 print('The answer: y = ',y_unknown[0][0])
13
14 print('...Double check...')
15
16 print('y = a*x + b')
17 print('a = ', imported_model.coef_[0][0])
18 print('b = ', imported_model.intercept_[0])
19
20 print('y = a*X + b')
21 print(imported_model.coef_[0][0]*X_unknown[0] + imported_model.intercept_[0])
```


Funkcja druga

1. Otwieramy plik **csv** i dodajemy na końcu parametry funkcji x i y
2. Przypisujemy do zmiennej w formie Data Frame zawartość zaktualizowanego pliku csv
3. Przekształcamy kolumny w łańcuchy numpy o odpowiednim kształcie
4. Przypisujemy do zmiennej model regresji liniowej
5. Dokonujemy treningu (fit) przy pomocy nowych danych (łańcuchów Numpy)
6. Zmodyfikowany model zapisujemy w pliku **pickle** o nazwie **our_model.pkl** (istniejący nadpisujemy)

Przydatny kod dla funkcji 2

```
1 import pandas as pd
2 import numpy as np
3 from sklearn.linear_model import LinearRegression
4 import pickle
5 |
6 x = 11
7 y = 21.65
8
9 # Add row:
10 # Read the file as DataFrame
11 df = pd.read_csv('10_points.csv')
12
13 # Add values as last row in DF (len(df.index))
14 df.loc[len(df.index)] = [x,y] # list
15
16 # Overwrite old file with new values (reset indexes)
17 df.to_csv(path2csv, index=False)
18
19
20 # Read data
21 df = pd.read_csv('10_points.csv')
22
23 # Reshape data for modelling
24 X = df['x'].values.reshape(-1,1) # values converts it into a numpy array
25 y = df['y'].values.reshape(-1,1) # -1 means that calculate the dimension of
   rows, but have 1 column
26
27 # Instantiate the model
28 our_model = LinearRegression()
29
30 # Fit the model
31
32 our_model.fit(X, y)
33
34 # Print coefficient and intercept
35 print('y = a*x + b')
36 print('a = ', our_model.coef_[0][0])
37 print('b = ', our_model.intercept_[0])
38
39 # Export the model
40 print('...Exporting the model...')
41 pickle.dump(our_model, open('our_model.pkl', 'wb'))
```

Dziękuję za uwagę

A thick, hand-drawn style orange line that underlines the text "Dziękuję za uwagę".