

Laboratorium 3 – Regresja, część II

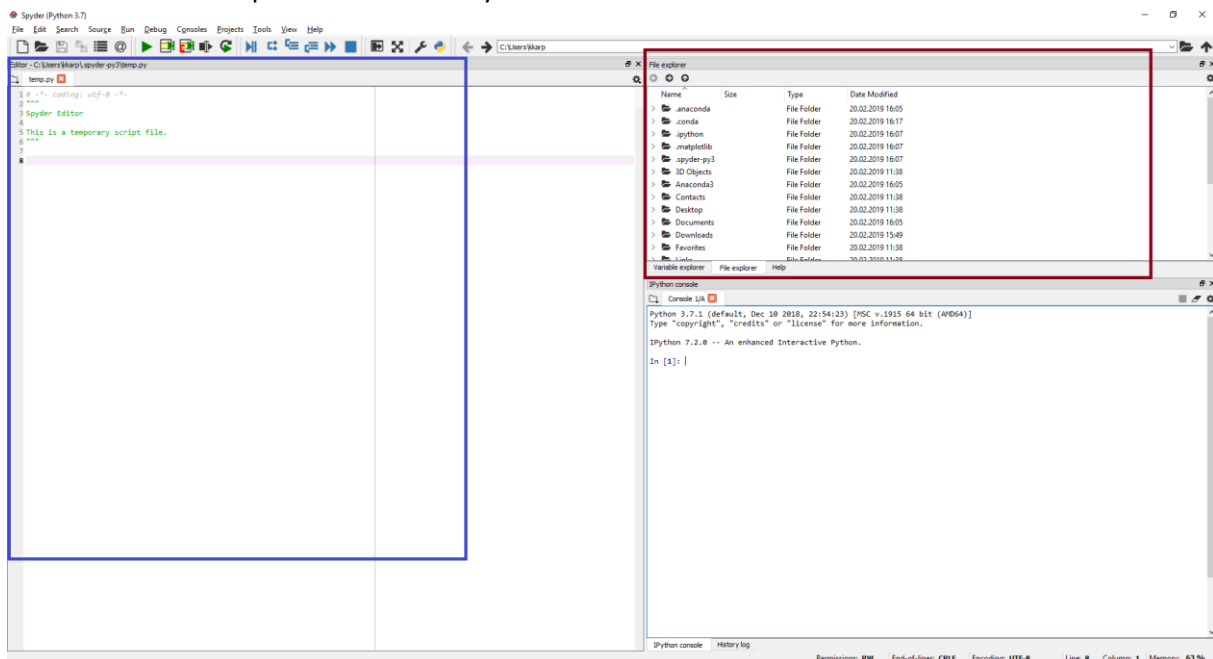
Zadania laboratoryjne:

1. Przygotowanie środowiska do pracy.

- 1.1. Pobrać i zainstalować platformę Anaconda (<https://www.anaconda.com/distribution/>).
- 1.2. Uruchomić platformę Anaconda, a następnie aplikację Spyder.
- 1.3. Na komputerze utworzyć nowy folder i zapisać w nim otrzymany od prowadzącego plik z danymi (Position_Salaries.csv).

UWAGA! Pliki mogą być wykorzystywane podczas kolejnych laboratoriów, dlatego po zakończeniu zajęć należy skopiować folder na prywatny nośnik.

- 1.4. W programie Spyder ustawić swój folder jako katalog roboczy (obszar zaznaczony czerwonym prostokątem na Rysunku 1.). Zapisać pod dowolną nazwą plik temp.py (widoczny w obszarze zaznaczonym niebieskim prostokątem na Rysunku 1.) w tym samym folderze co plik ze zbiorem danych.



Rysunek 1. Interfejs programu Spyder.

2. Regresja wielomianowa.

Wykorzystując napisany na poprzednich zajęciach skrypt, przygotować zbiór danych do analizy.

- 2.1. Wczytać zbiór danych Position_Salaries.csv.
- 2.2. Utworzyć macierz zmiennych niezależnych (Level) oraz wektor zmiennych zależnych (Salary).
- 2.3. Nie tworzyć zestawów treningowych i testowych. Jest to spowodowane charakterem danych.
- 2.4. Dopasować model regresji liniowej do zestawu danych wykorzystując klasę LinearRegression.
- 2.5. Wykreślić regresję liniową lin_reg.
- 2.6. Wykorzystać klasę PolynomialFeatures do utworzenia macierzy wielomianów.

```
from sklearn.preprocessing import PolynomialFeatures
poly_reg = PolynomialFeatures(degree = 2)
```

```
# poly_reg to obiekt będący narzędziem służącym do
transformacji macierzy X do macierzy wielomianów X_poly

X_poly = poly_reg.fit_transform(X)
poly_reg.fit(X_poly, y)
```

- 2.7. Wyświetlić macierz X_poly i przeanalizować jakie wartości się w niej znajdują.
2.8. Dopasować kolejny model regresji liniowej, tym razem do danych zawartych w macierzy X_poly.

```
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)
```

- 2.9. Wykreślić regresję liniową lin_reg2.

```
plt.scatter(X, y, color = 'red')
plt.plot(X_grid,
lin_reg_2.predict(poly_reg.fit_transform(X_grid)), color =
'blue')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

- 2.10. Wykorzystać utworzone modele do predykcji parametru Salary dla Level = 6.

```
arr=np.array([6])
arr=arr.reshape(1, -1)
lin_reg.predict(arr)

lin_reg_2.predict(poly_reg.fit_transform(6))
```

- 2.11. Zwiększyć stopień wielomianu z podpunktu 2.6 do degree = 3, ponownie uruchomić cały skrypt, wykreślić model tak jak w podpunkcie 2.9 oraz dokonać predykcji parametru tak jak w punkcie 2.10.

- 2.12. Zwiększyć stopień wielomianu z podpunktu 2.6 do degree = 4, ponownie uruchomić cały skrypt i wykreślić model tak jak w podpunkcie 2.9 oraz dokonać predykcji parametru tak jak w punkcie 2.10.

- 2.13. Wykonać sprawozdanie z realizacji podpunktów 2.1 – 2.12. Przestać w formie pliku pdf do serwisu moodle. Plik powinien zawierać następujące informacje:

- Wykres z podpunktu 2.5.
- Wykres z podpunktu 2.9.
- Wykres z podpunktu 2.11.
- Wykres z podpunktu 2.12.
- Wartości obliczone w punkcie 2.10.
- Odpowiedź na pytanie: Który z modeli dokonał najlepszej predykcji parametru Salary dla Level = 6? Czym to jest spowodowane?