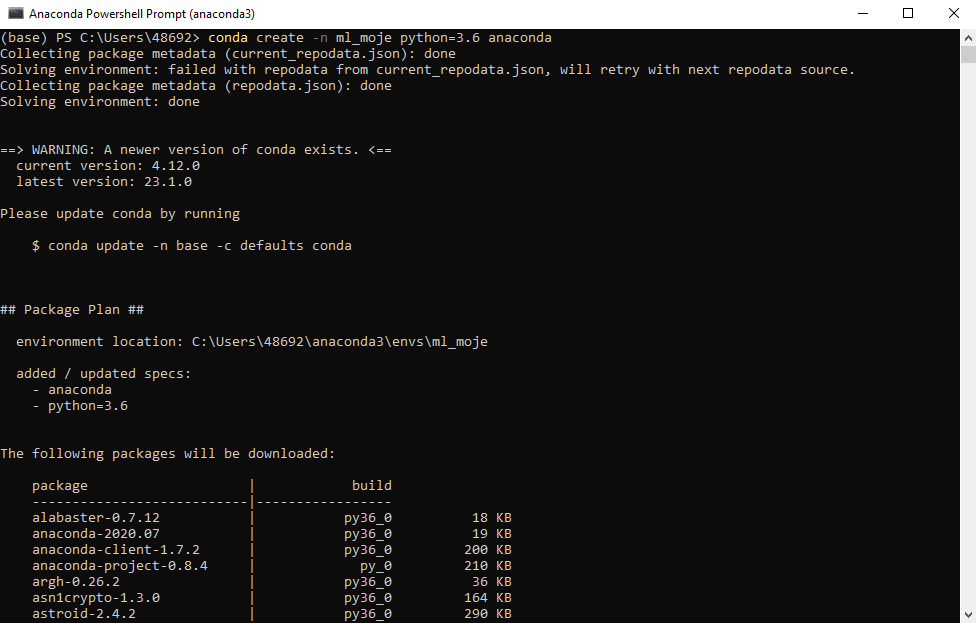
**LABORATORIUM 2**

ALEKSANDRA PEŁKA

**Ćwiczenie 1**

Utworzenie środowiska o nazwie ml\_moje:

**conda create -n ml\_moje python=3.6 anaconda**



Obraz zawierający tekst

Opis wygenerowany automatycznie

Aktywacja utworzonego środowiska i sprawdzenie, czy jest aktywne:

**conda activate ml\_moje**

**conda env list**

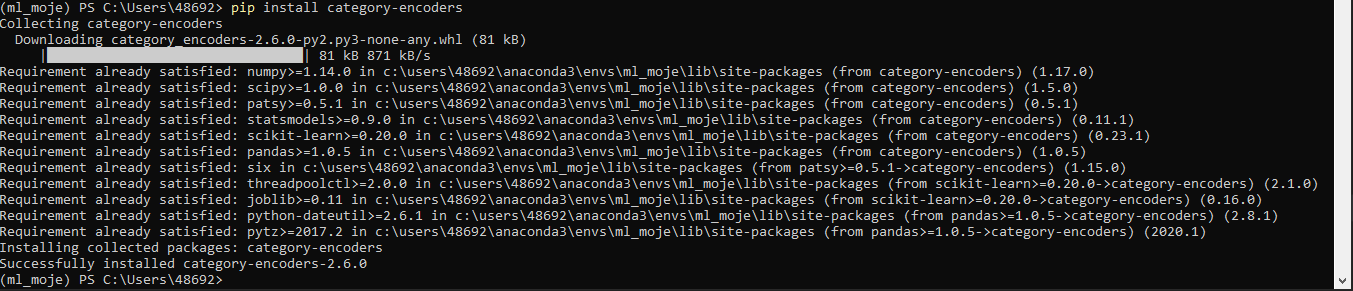
Obraz zawierający tekst

Opis wygenerowany automatycznie

**Instalacja potrzebnych bibliotek i ich opis:**

1. **pip install category-encoders**

Biblioteka zapewnia szereg funkcji umożliwiających przekształcenie zmiennych na dane typu numerycznego.



1. **pip install -c conda-forge feature\_engine**

Biblioteka posiadająca wiele przydatnych funkcji do wykonania odpowiedniego preprocessingu danych, przygotowania danych do dalszej analizy.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. **pip install featuretools**

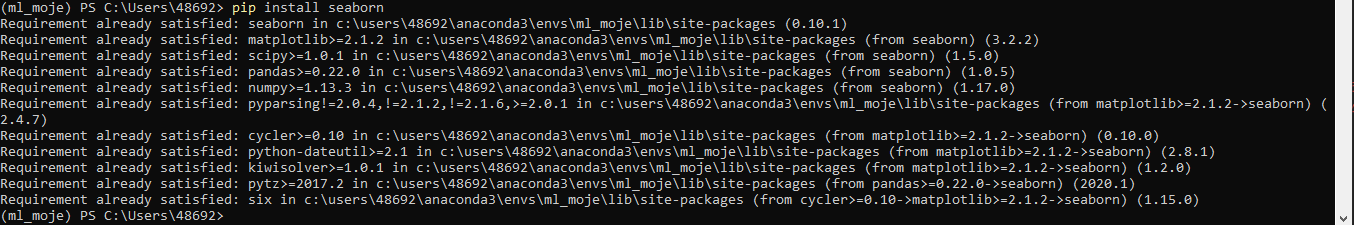
Framework umożliwiający zautomatyzowanie przetwarzania danych, ich odpowiednie przygotowanie, wybór najlepszych parametrów do modelu.

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. **pip install seaborn**

Biblioteka wykorzystywana w celu wizualizacji danych, zaprezentowania otrzymanych rezultatów w atrakcyjny sposób na różnego rodzaju wykresach.



1. **conda install numpy**

Biblioteka umożliwiająca przede wszystkim wykonywanie skomplikowanych operacji na tablicach wielowymiarowych i macierzach w łatwy i szybki sposób.

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. **conda install pandas**

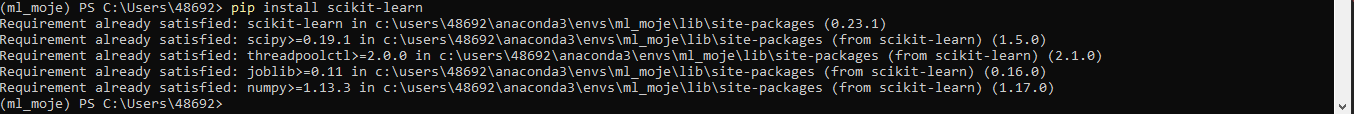
Podobnie jak w przypadku numpy, biblioteka pandas służy do wykonywania operacji na danych w postaci tabel, ale również na szeregach czasowych; ułatwia przygotowanie i analizę danych (wczytanie, czyszczenie danych, tworzenie wykresów, korelacji).

Obraz zawierający tekst

Opis wygenerowany automatycznie

1. **pip install scikit-learn**

Biblioteka umożliwiająca tworzenie modeli opartych na algorytmach wykorzystywanych w uczeniu maszynowym tj. regresji, klasyfikacji, klasteryzacji.



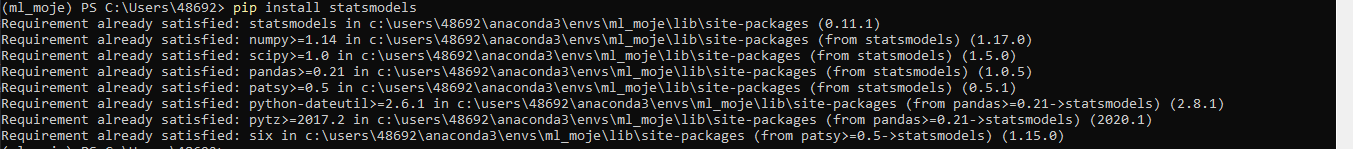
1. **pip install scipy**

Biblioteka przeznaczona do wykonywania złożonych obliczeń w optymalny sposób, zapewnia dostęp do wielu funkcji matematycznych i statystycznych.



1. **pip install statsmodels**

Biblioteka wykorzystywana m.in. w celu obliczania statystyk, tworzenia, dopasowania modeli statystycznych oraz sprawdzania ich jakości.



**Różnica między pip a conda:**

Za pomocą pip można jedynie instalować pakiety wewnątrz danego środowiska, natomiast polecenie conda umożliwia zarówno instalować potrzebne biblioteki wewnątrz środowiska, jak i tworzyć samo środowisko. Ponadto pip umożliwia instalacje jedynie bibliotek Pythona, a za pomocą condy można zainstalować również bibliotekę odpowiednią dla języka R.

**Stworzenie pliku z kopią środowiska w formacie yaml:**

Kopię środowiska wyeksportowaną do pliku .yml wykonano przy użyciu polecenia:

**conda env export > ml\_moje.yml**

Obraz zawierający tekst

Opis wygenerowany automatycznie

Utworzenie środowiska z wygenerowanego pliku ml\_moje.yml, zamieszczonego w repozytorium, zawierającego konfigurację środowiska i wszystkie zainstalowane pakiety:

**conda env create -f ml\_moje.yml**

Obraz zawierający tekst, zrzut ekranu, monitor, czarne

Opis wygenerowany automatycznie

**Jupyter notebook:**

Sprawdzenie poprawności dodania środowiska:

**conda info --envs**

Obraz zawierający tekst

Opis wygenerowany automatycznie

Aktywacja środowiska i sprawdzenie, czy zostało aktywowane:

**conda activate ml\_moje**

**conda info --envs**

Obraz zawierający tekst, w pomieszczeniu

Opis wygenerowany automatycznie

Instalacja biblioteki ipykernel:

**pip install ipykernel**

Obraz zawierający tekst

Opis wygenerowany automatycznie

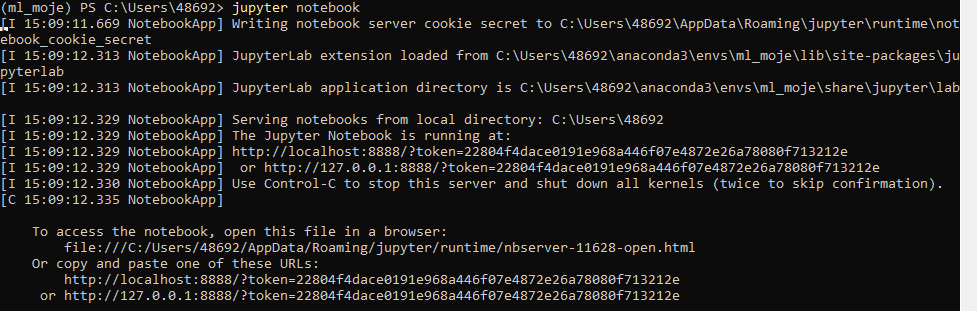
Dodanie środowiska wirtualnego z wybraną nazwą:

**python -m ipykernel install --user --name ml\_moje --display-name "Python py36"**



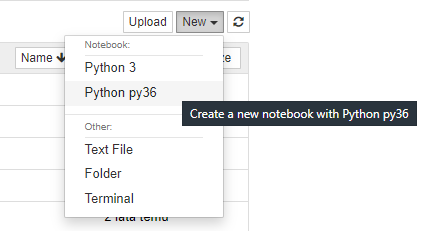
Uruchomienie Jupyter Notebook:

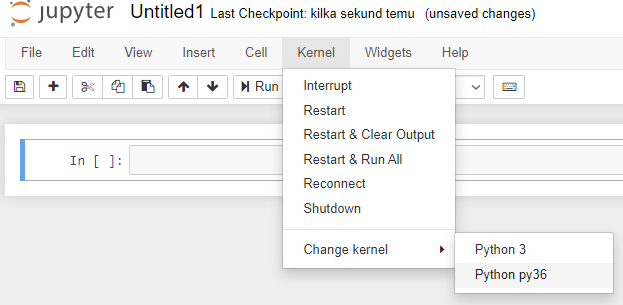
**jupyter notebook**



Wybór odpowiedniego środowiska w Jupyter Notebook:

**New > NazwaŚrodowiska** lub **Kernel > Change kernel > NazwaŚrodowiska**





**Co daje możliwość ustanowienia własnego środowiska:**

Ustanowienie własnego środowiska:

- umożliwia przechowywanie w danym środowisku tylko wybranych bibliotek potrzebnych do projektu

- zapewnia zgodność wersji pomiędzy wykorzystanymi bibliotekami (czasem potrzeba wykorzystania starszych wersji)

- daje możliwość posiadania różnych wersji tego samego pakietu w odrębnych środowiskach