

Laboratorium 2 – PANDAS

Streszczenie

Moduł "Pandas" jest zestawem narzędzi umożliwiającym sprawne manipulowanie zestawem danych. Rozszerza on możliwości PYTHON o łatwy import i eksport danych m.in. do plików w formatach tekstowych (csv) czy konkretnych aplikacji (excel). Dane są przechowywane w tabelach tzw. DataFrame. "Pandas" dostarcza wiele narzędzi do selekcji, łączenia i sortowania danych.

1 CEL

Celem głównym zadania jest zapoznanie się z podstawowymi funkcjami pakietu "PANDAS" oraz wykorzystanie wraz z nim innych narzędzi wizualizacji i analizy danych m.in: numpy, scipy, matplotlib, sklearn.

Ważniejsze funkcje: 'DataFrame, sort_index, sort_values, groupby, read..., write itp.'

2 Treść i zadania

UWAGA!!! Zajęcia prowadzone są po raz pierwszy. Wszelkie wątpliwości co do treści zadań i ich kolejności proszę zgłaszać do osoby przygotowującej zestaw:

Andrzej Łysko

alysko@zut.edu.pl

W celu konsultacji proszę się zgłosić do pokoju 303 WI2.

Wykonaj zadania i wyświetl wyniki. Kod zapisuj w jednym pliku do wglądu dla prowadzącego.

Sprawozdanie z zajęć zawierające wykorzystane fragmenty programu, wyniki działań itp. umieść na portalu elerning w zakładce 'PiAD/Laboratoria/Lab2'. elerning

Proszę się zalogować poświadczeniami używanymi w aplikacji 'edziekanat' i potwierdzić udział w kursie.

- Manipulowanie danymi:

Zapoznaj się z obiektem `'pandas.DataFrame'` i utwórz:

tabelę złożoną z liczb losowych przedziału normalnego złożoną z trzech kolumn z nagłówkiem (A,B,C) i pięciu wierszy z indeksem o nazwie "data" złożonym z dat w przedziale od 2020-03-01 do 2020-03-05 np.

data	A	B	C
2020-03-01	0,720184489	0,657752214	0,297794824
2020-03-02	0,775572034	0,532276741	0,051391112
2020-03-03	0,346071872	0,22057064	0,640917109
2020-03-04	0,475104139	0,228407696	0,507173083
2020-03-05	0,475104139	0,899298837	0,541077542

- Wygeneruj tabelę złożoną z liczb losowych i indeksie "id" w formacie 'integer' złożoną z 20 wierszy i trzech kolumn ('A','B','C'). Następnie:

- wybierz trzy pierwsze wiersze z tabeli,
- wybierz trzy ostatnie wiersze z tabeli,
- wyświetl nazwę indeksu tabeli,
- wyświetl nazwy kolumn,
- wyświetl tylko dane bez indeksów i nagłówków kolumn,
- wybierz pięć losowo wybranych wierszy,
- wybierz wartości kolumny 'A' a następnie 'A' i 'B',
- zapoznaj się z funkcją `'iloc'` i wyświetl:
trzy pierwsze wiersze i kolumny 'A' i 'B'.
wiersz piąty
wiersze 0,5,6,7 i kolumny 1 i 2

- Zapoznaj się z funkcją `'describe'` i wyświetl podstawowe statystyki tabeli:

- sprawdź które dane są większe od 0,
- wyświetl tylko dane większe od 0,
- wybierz kolumny A tylko dane większe od 0,
- policz średnią w kolumnach,
- policz średnią w wierszach.

- Zapoznaj się z funkcją `'concat'`. Utwórz dwie dowolne tabele i połącz je ze sobą.

Dokonaj transpozycji nowej tabeli.

- sortowanie: W tabelach DataFrame mogą być umieszczone różne typy danych:

```
df = pd.DataFrame("x": [1, 2, 3, 4, 5], "y": ['a', 'b', 'a', 'b', 'b'], index=np.arange(5)) df.index.name='id' print(df)
```

- posortuj dane po 'id' rosnąco,
- posortuj dane po kolumnie 'y' malejąco.

- Grupowanie danych (prześledź działania):

```
sloownik = 'Day': ['Mon', 'Tue', 'Mon', 'Tue', 'Mon'], 'Fruit': ['Apple', 'Apple', 'Banana', 'Banana', 'Apple'], 'Pound': [10, 15, 50, 40, 5], 'Profit': [20, 30, 25, 20, 10]
df3 = pd.DataFrame(sloownik)
print(df3)
print(df3.groupby('Day').sum())
print(df3.groupby(['Day', 'Fruit']).sum())
```

- Wypełnianie danych:

```
df=pd.DataFrame(np.random.randn(20, 3), index=np.arange(20), columns=['A','B','C'])
df.index.name='id'
print(df)
Wykonaj i opisz jak działają poniższe komendy:
df['B']=1
print(df)
df.iloc[1,2]=10
print(df)
df[df<0]=-df
print(df)
```

- Uzupełnianie danych. Wykonaj i opisz działanie poniższych komend:

```
df.iloc[[0, 3], 1] = np.nan
print(df)
df.fillna(0, inplace=True)
print(df)
df.iloc[[0, 3], 1] = np.nan
df=df.replace(to_replace=np.nan,value=-9999)
print(df)
df.iloc[[0, 3], 1] = np.nan
print(pd.isnull(df))
```

Zadania:

```
df = pd.DataFrame("x": [1, 2, 3, 4, 5], 'y': ['a', 'b', 'a', 'b', 'b'])
```

1. Zgrupować tabele po zmiennej symbolicznej *X* , a następnie wyznaczyć średnią wartość atrybutu numerycznego *Y* w grupach wyznaczonych przez *X*,
2. Wyznaczyć rozkład licznosci atrybutów (*value_counts*).
3. Wczytać dane *autos.csv*, za pomocą polecenia *np.loadtxt* oraz *pandas.read_csv*. Sprawdź różnice.
4. Zgrupować ramkę danych po zmiennej *'make'* a następnie wyznaczyć średnie zużycie paliwa dla każdego z producentów.
5. Zgrupować ramkę danych po zmiennej *make* licznosci dla atrybutu *fuel-type*.
6. Dopasować wielomian 1 i 2 stopnia prognozujący wartość zmiennej *city-mpg*, względem *length* (*np.polyfit* ,*np.polyval*).
7. Wyznaczyć współczynnik korelacji pomiędzy zmiennymi (*scipy.stats*).
8. Zwizualizować wynik zaznaczając próbki oraz dopasowanie krzywą na tle próbek.
9. Dla zmiennej *length* utworzyć jednowymiarowy estymator funkcji gęstości
Użyj:
scipy.stats.gaussian_kde , proszę zwizualizować wynik przedstawiając jednocześnie próbki i funkcję gęstości. Do wykresu dodać legendę.
Użyj: (*plot(..., label='...')*, *legend*)
10. Utworzyć w jednym oknie graficznym dwa wykresy *ax=subplot(...)*, *ax.plot(...)*. Na drugim wykresie przedstawić analogicznie rozkład dla zmiennej *'width'*.
11. Utworzyć dwuwymiarowy estymator funkcji gęstości dla zmiennych *width* i *length* , wynik przedstawić graficznie w nowym oknie rysując próbki poleceniem *plot* oraz funkcję gęstości używając polecenia *meshgrid* i *contour*.
Wynik zapisać do plików w formacie *png* i *pdf* (*savefig*)

Korzystano m.in.:

<https://pandas.pydata.org/>

<https://github.com/yongtwang/engineering-python>