Advanced Institute for Artificial Intelligence

Pandas

Advanced Institute for Artificial Intelligence

https://advancedinstitute.ai

Sumário

- Pandas
- Estruturas
- Operações

Pandas

- Biblioteca Python de código aberto para análise de dados
- Pandas é composto por dois tipos de estruturas de dados:
  - Series
  - DataFrame

Pandas

- Uma série é um objeto unidimensional semelhante a uma matriz, lista ou coluna em uma tabela.
- Cada item possui um índice, por padrão o índice é de 0 a N
- Alternativamente, você pode especificar um índice para a série.

Pandas

- O construtor Series também pode converter um dictonário, usando as chaves do dicionário como índice.
- Você pode usar o índice para selecionar itens específicos da série
- É possível utilizar uma condição, que ao ser aplicada ao índice, retorna os elementos que atendem a condição
- Operações matemáticas podem ser feitas usando escalares e funções.

Pandas

DataFrame

- Estrutura de dados tablular composta de linhas e colunas
- planilha, tabela de banco de dados, etc
- Um Dataframe pode ser também um grupo de objetos series

Pandas

Montando um Dataframe

- Para criar um DataFrame a partir de estruturas de dados Python comuns, podemos passar um dicionário de listas para o construtor DataFrame.
- Parâmetro column permite indicar a ordem das colunas

Pandas

Montando um Dataframe

- Para criar um DataFrame a partir de estruturas de dados Python comuns, podemos passar um dicionário de listas para o construtor DataFrame.
- Parâmetro column permite indicar a ordem das colunas
- É possível criar um Dataframe a partir de um CSV
  - fromcsv = pd.read_csv('teste.csv')
- É possível criar um Dataframe a partir de uma URL
  - from_url = pd.read_table(url)

Pandas

- Info() : Retorna informações diversas
  - número de linhas, colunas, quantidade de espaço ocupado e tipo de dado
- Describe() : Retorna informações estatísticas a respeito dos dados de cada coluna

Pandas

- Definindo uma coluna do dataframe como índice
- users.set_index('user_id', inplace = True)
- inplace=true indica que a própria referência deve ser alterada, senão um novo dataframe é retornado

Pandas

Seleção de linhas e colunas

- A seleção de uma coluna retorna um objeto series
  - display(from_url[['rank', 'price']].head())
  - print(from_url[['rank', 'price']].head())
- Selecionando linhas
  - display(users[users.age $>$ 25].head(3))

Pandas

Seleção de linhas e colunas loc, iloc e ix:

- .loc [] funciona nos rótulos do seu índice. Isso significa que, se você fornecer o loc [2], procurará os valores do seu DataFrame com um índice rotulado 2.
- .iloc [] trabalha nas posições em seu índice. Isso significa que, se você fornecer iloc [2], procurará os valores do seu DataFrame que estão no índice '2'.
- .ix [] é um caso mais complexo: quando o índice é baseado em números inteiros, você passa um rótulo para .ix [].

Pandas

Exemplos

```
df = pd.DataFrame(data=np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]]),
index= [2, 'A', 4], columns=[48, 49, 50])

# Pass '2' to 'loc'
print(df.loc[2])

# Pass '2' to 'iloc'
print(df.iloc[2])

# Pass '2' to 'ix'
print(df.ix[2])
```

Pandas

Adicionando colunas

- basta utilizar um nome de coluna inexistente e passar um objeto series como parâmetro

Table: Operadores Python e métodos Pandas

Operador Python	Método Pandas
+	add()
-	sub() , subtract()
*	mul() , multiply()
/	truediv() , div() , divide()
//	floordiv()
%	mod()
**	pow()

Pandas

Representação de dados ausentes:

- none: indica uma ausência de valor, a variável aponta para local nulo
- NaN (sigla para Not a Number): padrão IEEE que representa um número inválido
- Ao criar um dataframe usando algum tipo numérico (float, int, ...) todo objeto do tipo None é covertido automaticamente para NaN

```
vals1 = np.array([1, None, 3, 4])
vals1.sum()
vals2 = np.array([1, np.nan, 3, 4])
vals2.sum()
```

Pandas

Manipulando valores nulos

- isnull(): Gere uma máscara booleana indicando valores ausentes
- notnull(): Oposto de isnulll()
- fillna(): Retorna um dataframe filtrada sem dados nulos
- dropna(): Retorna um dataframe com valores ausentes preenchidos

Pandas

Índices podem ser compostos por múltiplos valores

```
index = [('California', 2000), ('California', 2010),
('New York', 2000), ('New York', 2010),
('Texas', 2000), ('Texas', 2010)]

populations = [33871648, 37253956,
18976457, 19378102,
20851820, 25145561]

pop = pd.Series(populations, index=index)
```

Pandas

Filtrar por um índice multivalorado tornar-se complexo

```
pop[[i for i in pop.index if i[1] == 2010]]
```

Método reindex permite criar novos índices a partir de índices multivalorados

```
pop = pop.reindex(index)
print(pop)
pop[:, 2010]
```

Concatenação de dataframes: método concat

```
ser1 = pd.Series(['A', 'B', 'C'], index=[1, 2, 3])
ser2 = pd.Series(['D', 'E', 'F'], index=[4, 5, 6])
pd.concat([ser1, ser2])
```

```
método verify_integrity = True verifica a integridade dos índices
método ignore_index=True -> refaz o índice, ignorando o que havia anteriorment
```

Pandas

Join : concatena dois dataframes, usando um conjunto de colunas como critério

No primeiro caso as colunas que não são comuns, são preenchidas com valores nulos

```
df5 = make_df('ABC', [1, 2])
df6 = make_df('BCD', [3, 4])
print(df5); print(df6); print(pd.concat([df5, df6])
```

Nesse caso as colunas que não estão presentes em df5 são descartadas

```
print(df5); print(df6);
print(pd.concat([df5, df6], join_axes=[df5.columns]))
```

Pandas

merge: implementa algebra relacionao do tipo

- 1 para 1
- 1 para N
- M para N

1 para 1

```
df1 = pd.DataFrame({'employee': ['Bob', 'Jake', 'Lisa', 'Sue'],
'group': ['Accounting', 'Engineering', 'Engineering', 'HR']})
df2 = pd.DataFrame({'employee': ['Lisa', 'Bob', 'Jake', 'Sue'],
'hire_date': [2004, 2008, 2012, 2014]})
print(df1); print(df2)
df3 = pd.merge(df1, df2)
df3
```

Pandas

1 para muitos

```
df4 = pd.DataFrame({'group': ['Accounting', 'Engineering', 'HR'],
'supervisor': ['Carly', 'Guido', 'Steve']})
print(df3); print(df4); print(pd.merge(df3, df4))
```

muitos para muitos

```
df5 = pd.DataFrame({'group': ['Accounting', 'Accounting',
'Engineering', 'Engineering', 'HR', 'HR'],'skills': ['math', 'spreadsheets',
'spreadsheets', 'organization']})
print(df1); print(df5); print(pd.merge(df1, df5))
```

Pandas

Outros parâmetros para merge

- on: define a coluna que vai ser usada como critério para o merge
- left_on e right_on: define as colunas de cada tabela, caso tenham nomes distintos
- how : define aritmética do join
  - inner : retorna apenas a intersecção
  - outer : linhas fora da intersecção, retornam com valores nulos
  - left : linhas fora da intersecção no primeiro dataframe retornam com valores nulos e as linhas do segundo dataframe fora da intersecção são descartadas
  - right: linhas fora da intersecção no primeiro dataframe são descartadas e as linhas do segundo dataframe fora da intersecção retornam com valores nulos