Lecture 7: Python Pandas Library, Python Classes

CS 167: Machine Learning

Pandas

Pandas is a Python data analysis library.

- built on top of another library called NumPy

Run the following import statement to see if you have it properly installed:

```
import pandas
```

If not go here to get it: http://pandas.pydata.org/getpandas.html

Try it out

Read in a data set from a csv file and then print it out:

```
import pandas
data = pandas.read_csv('restaurant.csv')
print(data)
```

What does this do?

```
print(data['price'])
```

How about this?

```
print(data[3])
```


Selecting Data

Try these one at a time - each in its own cell:

```
print(data.loc[3])
print(data.loc[3,'type'])
print(data.iloc[3])
print(data.iloc[3,8])
```

Look for the *Name* in each of these:

```
print(data['est'])
print(data.loc[5])
```

What is the difference between `iloc` and `loc`?

What is this doing?

```
data['type'][5]
```

What are the types of all these things?

Try these, discuss the types you're discovering:

```
print(type(data))
print(type(data['est']))
print(type(data.loc[0]))
print(type(data.iloc[3,8]))
```

Exercises/Questions:

- **Does the data always come in as strings or does it infer the right type?** Read in the iris data set and check if it brings them in as numbers.
- **What does it do with missing values?** Read in the Titantic data set and find out.

Slicing in Dataframes

Try more:

```
print(data.iloc[3:6,4:8])
print(data['rain'].iloc[4])
print(data[ [True,False,False,True,True,False,True,True,\
             False,False,False,True] ])
```

Explain what you see

It's getting crazy...

What in the world is going on here?

```
print(data['type'] == 'Thai')

print(data[ data['type'] == 'Thai' ])
```

Exercise: *Print out the rows of the iris data with petal length greater than 6.*

Revisiting our data prep tasks

Let's open up the Titanic data set and try out a few Pandas functions:

```
titanic_data = pandas.read_csv('titanicFull.csv')
print(titanic_data['Sex'].unique())
mean_age =  titanic_data['Age'].mean()
print(mean_age)
titanic_data['Age'] = titanic_data['Age'].fillna( mean_age )
print(titanic_data)
print(titanic_data[titanic_data['Embarked'].isnull()])
```

Discuss: **What do `unique()`, `mean()`, `isnull()`, and `fillna()` do?**

Exercise: *Find the median sepal width from the iris data set*

Arithmetic on Columns

```
iris_data = pandas.read_csv('irisData.csv')

avg_thing_length = (iris_data['petal length'] \
                  + iris_data['sepal length'])/2

print(avg_thing_length)
iris_data['new column'] = avg_thing_length
print(iris_data)
```

Discuss: What does + do to Pandas series? Does it do the same thing it does to Python lists?

Exercises

Internet Research Question: **Can I quickly find the standard deviation of a Pandas column?**

Exercise: *Normalize the Age column of the Titanic data set using the Z-Score in one line of code.*

Recall: the Z-score for x_i from a sequence of values $x_1, x_2, \ldots, x_n$ is

$$\frac{x_i - \mu}{\sigma}$$

where μ is the mean of $x_1, x_2, \ldots, x_n$ and σ is the standard deviation

Python Classes: First Pass

```
class RectangleV1:
    x = 0
    y = 0
    def area(self):
        return self.x * self.y


one_rect = RectangleV1()
another_rect = RectangleV1()
one_rect.x = 10
one_rect.y = 4
print(one_rect.area())
print(another_rect.area())
```

```
40
0
```


Constructor

```
class RectangleV2:

    def __init__(self):
        self.x = 0
        self.y = 0

    def area(self):
        return self.x * self.y


one_rect = RectangleV2()
another_rect = RectangleV2()
one_rect.x = 10
one_rect.y = 4
print(one_rect.area())
print(another_rect.area())
```

```
40
0
```

Default Arguments

```
class RectangleV3:

    def __init__(self,startx=0,starty=0):
        self.x = startx
        self.y = starty

    def area(self):
        return self.x * self.y


one_rect = RectangleV3()
another_rect = RectangleV3(8,13)
one_rect.x = 10
one_rect.y = 4
print(one_rect.area())
print(another_rect.area())
```

```
40
104
```


Hiding Attrubutes

```
class RectangleV4:

    def __init__(self,startx=0,starty=0):
        self.__x = startx
        self.__y = starty

    def area(self):
        return self.__x * self.__y


one_rect = RectangleV4()
another_rect = RectangleV4(8,13)
one_rect.__x = 10
one_rect.__y = 4
print(one_rect.area())
print(another_rect.area())
```

```
0
104
```