Social Data Science: Machine Learning & Econometrics

Exercise class 3

March 5, 2020

Todays quick warmup

Q: Today a bit different from the last times. We will have a quiz on tracebacks and simple mistakes!

Error #1

Q: What is wrong?

```
---------------------------------------------------
TypeError                 Traceback (most recent call last)
<ipython-input-2-28f4dd90f3ff> in <module>
      2     return x + y
      3
----> 4 add('a', 3)

<ipython-input-2-28f4dd90f3ff> in add(x, y)
      1 def add(x,y):
----> 2     return x + y
      3
      4 add('a', 3)

TypeError: can only concatenate str (not "int") to str
```

Error #1

Q: What is wrong?

```
---------------------------------------------------
TypeError          Traceback (most recent call last)
<ipython-input-2-28f4dd90f3ff> in <module>
      2     return x + y
      3
----> 4 add('a', 3)

<ipython-input-2-28f4dd90f3ff> in add(x, y)
      1 def add(x,y):
----> 2     return x + y
      3
      4 add('a', 3)

TypeError: can only concatenate str (not "int") to str
```

- Integers and strings dont add: `add('a', 3)`
- This fails exactly when the function tries to add. *There is no type checking!*

Error #2

Q: What is wrong? How do you do this in a way that doesn't fail?

```
---------------------------------------------------------
KeyError                        Traceback (most recent call last)
<ipython-input-14-758aff63242d> in <module>
----> 1 {'a':1, 'b':2,'c':3}['d']

KeyError: 'd'
```

Error #2

Q: What is wrong? How do you do this in a way that doesn't fail?

```
---------------------------------------------------------
KeyError                        Traceback (most recent call last)
<ipython-input-14-758aff63242d> in <module>
----> 1 {'a':1, 'b':2,'c':3}['d']

KeyError: 'd'
```

- `{'a':1, 'b':2,'c':3}` doesn't have a `d` key.
- Either `{'a':1, 'b':2,'c':3}.get('d')` (returns `None`), or

```
try:
    {'a':1, 'b':2,'c':3}['d']
except KeyError:
    # do something
```

Error #3

Q_1: How do you read a traceback? Top-to-bottom, or bottom-to-top?
Q_2: What do you do if you don't understand the traceback after reading it?
Q_3: Could you do (2) before attempting (1)?
Q_4: Why is the middle part of a traceback often useless to read?

Error #3

Q_1: How do you read a traceback? Top-to-bottom, or bottom-to-top?
Q_2: What do you do if you don't understand the traceback after reading it?
Q_3: Could you do (2) before attempting (1)?
Q_4: Why is the middle part of a traceback often useless to read?

- Bottom-to-top!
- Paste it into google.
- Yes!
- This is often library-code that is doing error handling.

Error #3

Q: Finally produce code that raises the following errors

- `SyntaxError`
- `NameError`
- `ValueError`
- `ZeroDivisionError`
- `FileNotFoundError`

Last lecture in a nutshell

Potential outcomes framework (for causal inference):

- Each unit has associated with every value $D_i \in \{0, 1\}$ an outcome $Y_i(D_i)$.
- We only observe one version of the world so $D_i = 0$ xor $D_i = 1$ $\forall i$.
- We want to estimate $E[Y_i|D_i = 1] - E[Y_i|D_i = 0]$, but subtracting means gives

$$E[Y_i(1) - Y_i(0)|D_i = 1] + E[Y_i(0)|D_i = 1] - E[Y_i(0)|D_i = 0]$$

- Standard solution: (quasi)-randomization (**NOT** controlling for observables! (why?))

Note: we will use (quasi)-randomization throughout, so standard assumptions apply and must be fulfilled.

Last lecture in a nutshell

What we will be dealing with:

- (PS)Matching is one approach to this; basic idea is to compare apples to apples only (or maybe pears, but surely not bananas)
  - Intuitively this improves precision of the ATE
  - But importantly also a way to get local treatment effects.
- Methods to estimate the *treatment effect* locally (i.e. dropping the A in ATE).
  - who and how should we "match" to get consistent estimates of $Y(1) - Y(0)$ on the individual level?

Last lecture in a nutshell

Honest trees: step 1 is to understand honesty; no standard errors $\Rightarrow$ no causal inference.

- Imagine using a decision tree to "group similar observations"; then compute local $\hat{\tau}(\ell(x)) = \hat{\mu}_{D_i=1}^{\ell(x)} - \hat{\mu}_{D_i=0}^{\ell(x)}$ where $\ell(x)$ is the chosen leaf given x.
- Issue now is that grouping on standard CART criterion (1) groups "spuriously similar observations" $\rightarrow$ downwards bias on in-leaf variance.

$$Q^{CART}(\pi) = -\mathbb{E}_{\mathcal{S}^{te},\mathcal{S}^{tr}}[MSE_{\mu}(\mathcal{S}^{te},\mathcal{S}^{tr},\pi(\mathcal{S}^{tr}))] \tag{1}$$

- Solution? Estimate in-leaf variance using a separate estimation dataset $\mathcal{S}^{est}$. Take this step into account in the tree-building phase by modifying $Q(\pi)$ to

$$Q^{H}(\pi) = -\mathbb{E}_{\mathcal{S}^{te},\mathcal{S}^{est},\mathcal{S}^{tr}}[MSE_{\mu}(\mathcal{S}^{te},\mathcal{S}^{est},\pi(\mathcal{S}^{tr}))] \tag{2}$$

To get treatment effects, produce an unbiased estimator of MSE_{τ}