Hardware, Software, and Data Management

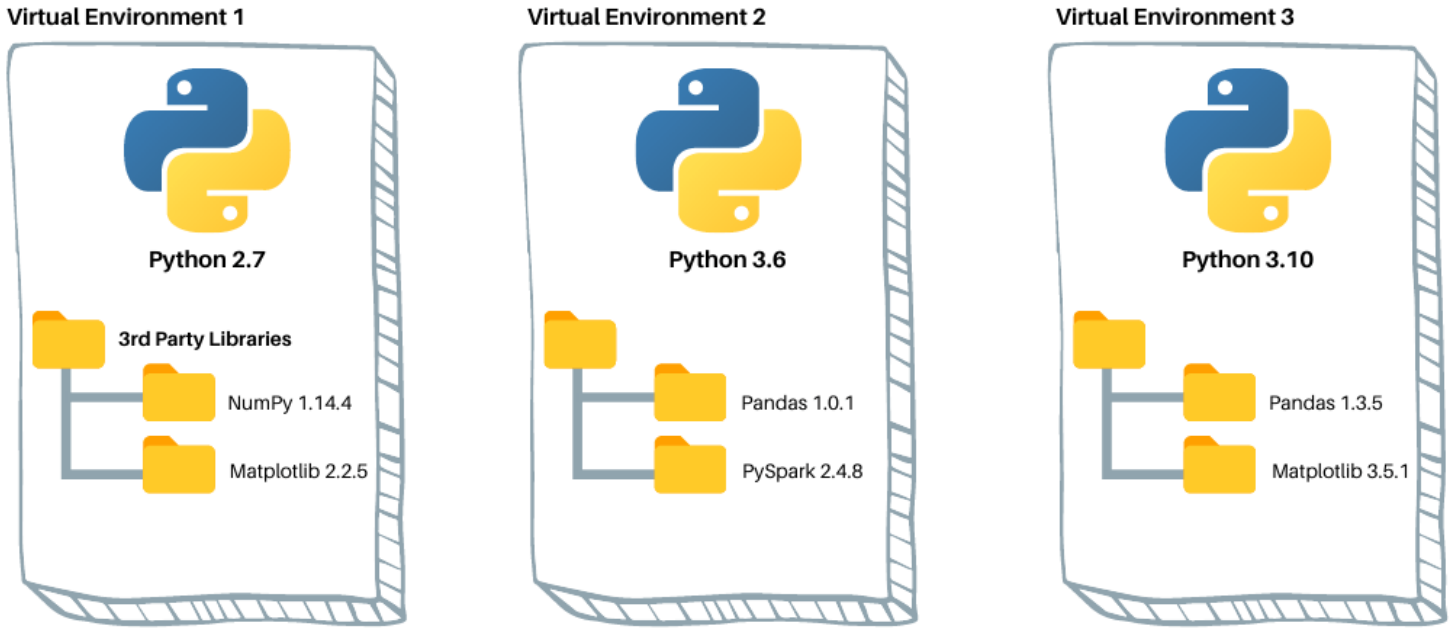
Tasks for This Week

- Understanding computing hardware
- Organizing and managing projects
- Managing environments
- Working with Jupyter lab/notebooks
- Working in Google Colab
- Introduction to SQL

File Management

Virtual Environments

Virtual Environments



dataquest.io

Virtual Environments

1. Clean

Virtual Environments

1. Clean
2. Dependency conflicts

Virtual Environments

1. Clean
2. Dependency conflicts
3. Reproducibility

Let’s set one up!

Getting started

As a rule of thumb, avoid installing to your base environment!

Windows

- Open Anaconda Prompt
- Dir: list files
- cd: change directory

macOS

- Open the terminal
- ls: list files
- cd: change directory

Linux

- Open the terminal
- ls: list files
- cd: change directory

Add Conda Forge

conda config -add channels conda-forge

conda config -set channel_priority strict

Creating an Environment

conda create -n ENVNAME python=x.x

conda activate ENVNAME

conda deactivate ENVNAME

conda install PACKAGE

conda list

conda env export > ENVIRONMENT.yml

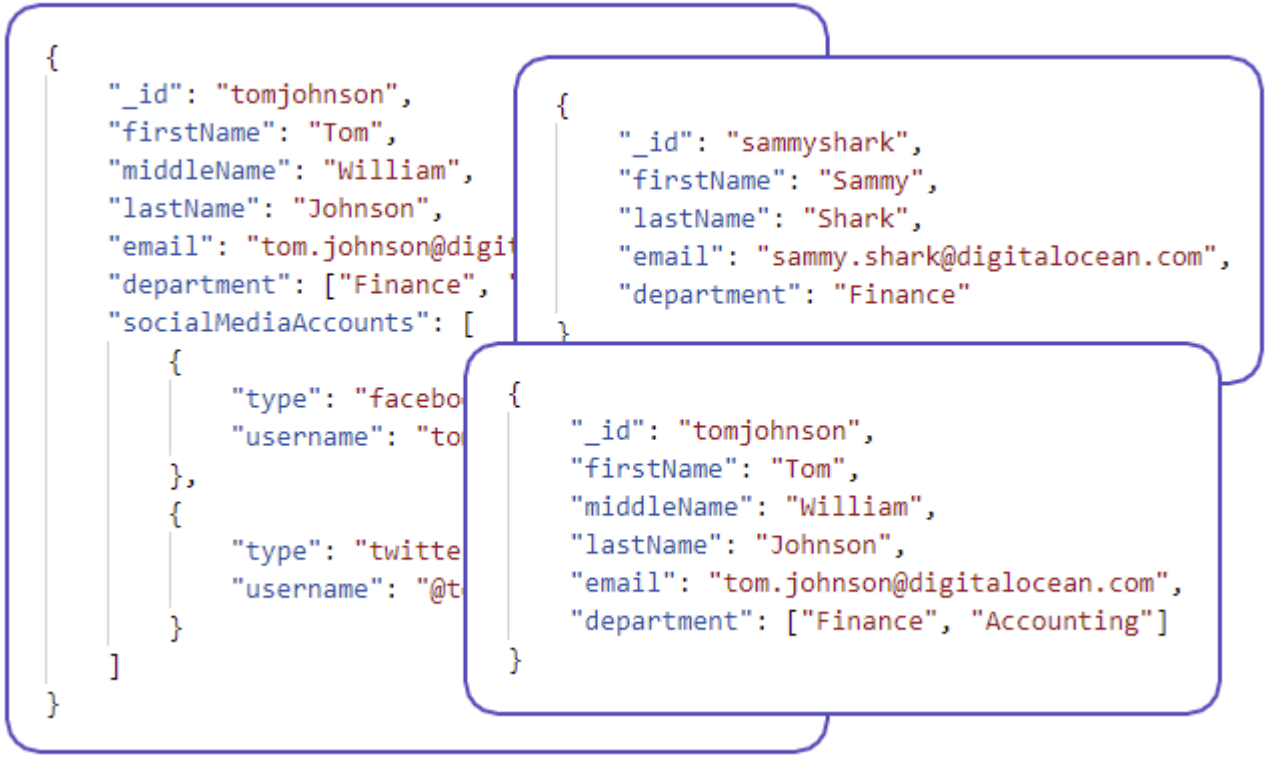
Conda env create -n ENVNAME -file ENVIRONMENT.yml

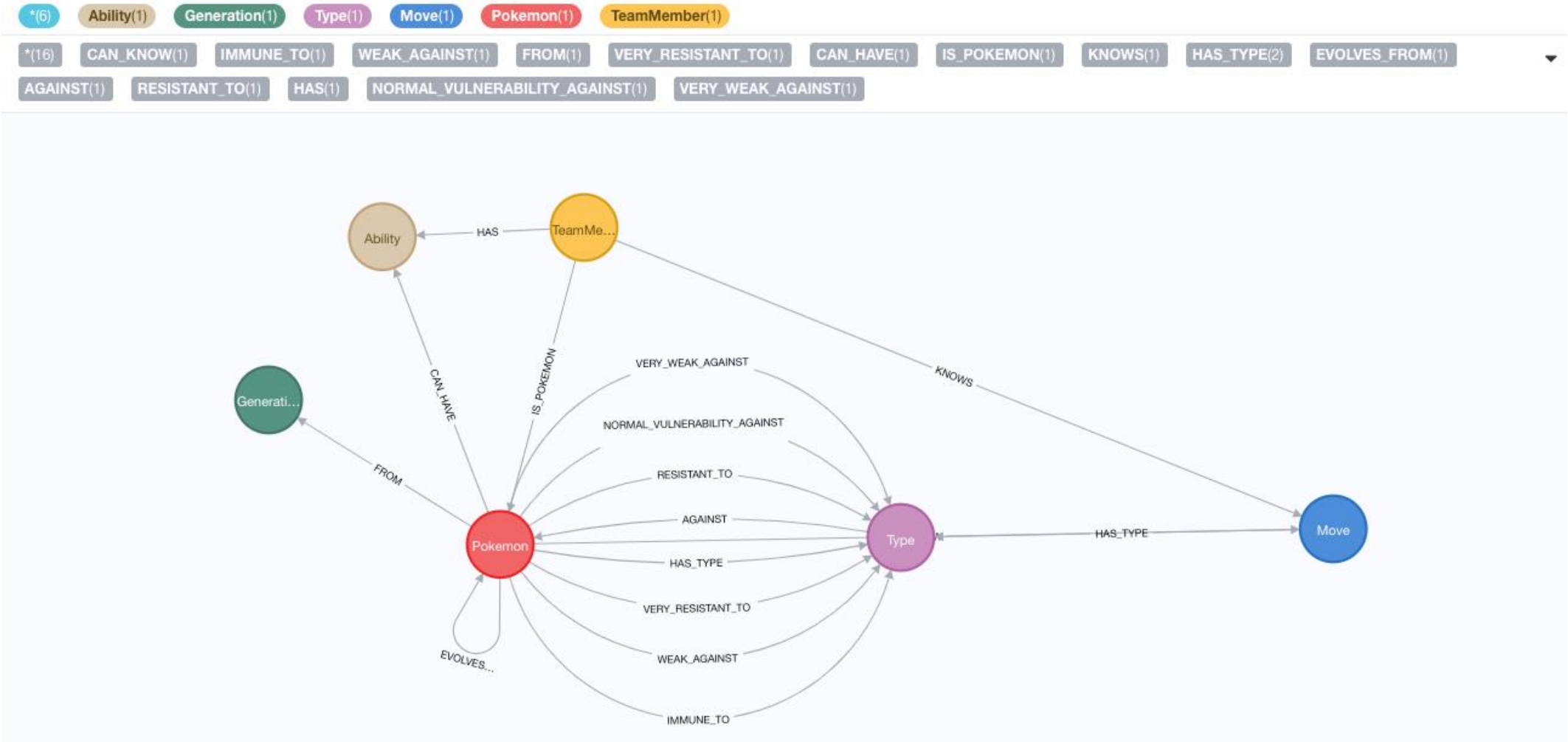
Create a new environment with the following: Python 3.8, jupyterlab, pandas, matplotlib, and seaborn

A Brief Introduction to SQL (and other databases)

Types of databases

- SQL/Relational database
- NoSQL:
  - Document
  - Graph
  - Key-value


{
"_id": "tomjohnson",
"firstName": "Tom",
"middleName": "William",
"lastName": "Johnson",
"email": "tom.johnson@digi
"department": ["Finance",
"socialMediaAccounts": [
{
"type": "facebo
"username": "to
},
{
"type": "twitte
"username": "@t
}
]
}
{
"_id": "sammyshark",
"firstName": "Sammy",
"lastName": "Shark",
"email": "sammy.shark@digitalocean.com",
"department": "Finance"
}
{
"_id": "tomjohnson",
"firstName": "Tom",
"middleName": "William",
"lastName": "Johnson",
"email": "tom.johnson@digitalocean.com",
"department": ["Finance", "Accounting"]
}


*(6)
Ability(1)
Generation(1)
Type(1)
Move(1)
Pokemon(1)
TeamMember(1)
*(16)
CAN_KNOW(1)
IMMUNE_TO(1)
WEAK_AGAINST(1)
FROM(1)
VERY_RESISTANT_TO(1)
CAN_HAVE(1)
IS_POKEMON(1)
KNOWS(1)
HAS_TYPE(2)
EVOLVES_FROM(1)
AGAINST(1)
RESISTANT_TO(1)
HAS(1)
NORMAL_VULNERABILITY_AGAINST(1)
VERY_WEAK_AGAINST(1)
Ability
HAS
TeamMe...
KNOWS
CAN_HAVE
IS_POKEMON
VERY_WEAK_AGAINST
Generati...
NORMAL_VULNERABILITY_AGAINST
RESISTANT_TO
FROM
AGAINST
Pokemon
Type
HAS_TYPE
Move
HAS_TYPE
VERY_RESISTANT_TO
EVOLVES...
WEAK_AGAINST
IMMUNE_TO

Key	Value
K1	AAA,BBB,CCC
K2	AAA,BBB
K3	AAA,DDD
K4	AAA,2,01/01/2015
K5	3,ZZZ,5623

LEFT JOIN

Everything on the left
+
anything on the right that matches

```
SELECT *
FROM TABLE_1
LEFT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

ANTI LEFT JOIN

Everything on the left
that is NOT on the right

```
SELECT *
FROM TABLE_1
LEFT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
WHERE TABLE_2.KEY IS NULL
```

RIGHT JOIN

Everything on the right
+
anything on the left that matches

```
SELECT *
FROM TABLE_1
RIGHT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

ANTI RIGHT JOIN

Everything on the right
that is NOT on the left

```
SELECT *
FROM TABLE_1
RIGHT JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
WHERE TABLE_1.KEY IS NULL
```

OUTER JOIN

Everything on the right
+
Everything on the left

```
SELECT *
FROM TABLE_1
OUTER JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

ANTI OUTER JOIN

Everything on the left and right
that is unique to each side

```
SELECT *
FROM TABLE_1
OUTER JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
WHERE TABLE_1.KEY IS NULL
OR TABLE_2.KEY IS NULL
```

INNER JOIN

Only the things that match on the
left AND the right

```
SELECT *
FROM TABLE_1
INNER JOIN TABLE_2
ON TABLE_1.KEY = TABLE_2.KEY
```

CROSS JOIN

All combination of rows from the
right and the left (cartesean
product)

```
SELECT *
FROM TABLE_1
CROSS JOIN TABLE_2
```

Assignment # 2

1. Create a conda/virtual environment for your project. Install the basic packages you think you will need (e.g. Pandas).
2. Create a github repository for your project that includes environment and readme files.
3. Add two files to your repo, one that uses your data for inferential modeling and one with a predictive model.
   - You must use two different models.
   - At least one file must be a jupyter notebook and use Python.
4. If possible, include your data in the repo.

Vocab

- Virtual/Conda/Computing environment
- Dependency
- RAM
- CPU
- GPU
- Hard drive/storage