# Better Know Your IP

# Example Application: Cybersecurity Analysis

# Example Application: Cybersecurity Analysis

- General Workflow: ETL

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics'

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers'

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'
- Push Button Time Allocation within Data Scientists:

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'
- Push Button Time Allocation within Data Scientists:
  - Analytics ($\sim$ 25%)
  - Deep Dive ($\sim$ 75%)

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'
- Push Button Time Allocation within Data Scientists:
  - Analytics ($\sim$ 25%)
  - Deep Dive ($\sim$ 75%)

A ~~Deep Dive~~ Dip Into Deep Dive

## Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'
- Push Button Time Allocation within Data Scientists:
  - Analytics ($\sim$ 25%)
  - Deep Dive ($\sim$ 75%)

### A ~~Deep Dive~~ Dip Into Deep Dive

- Manual sifting of small data

## Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'
- Push Button Time Allocation within Data Scientists:
  - Analytics ($\sim$ 25%)
  - Deep Dive ($\sim$ 75%)

A ~~Deep Dive~~ Dip Into Deep Dive

- Manual sifting of small data
- Find 'interesting' outliers

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'
- Push Button Time Allocation within Data Scientists:
  - Analytics ($\sim$ 25%)
  - Deep Dive ($\sim$ 75%)

A ~~Deep Dive~~ Dip Into Deep Dive

- Manual sifting of small data
- Find 'interesting' outliers
- Tell story about those outliers

# Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ 'Analytics' $\rightsquigarrow$ 'Outliers' $\rightsquigarrow$ 'Deep Dive'
- Push Button Time Allocation within Data Scientists:
  - Analytics ($\sim$ 25%)
  - Deep Dive ($\sim$ 75%)

## A ~~Deep Dive~~ Dip Into Deep Dive

- Manual sifting of small data
- Find 'interesting' outliers
- Tell story about those outliers
- Including, why the outlier was flagged as such

## Example Application: Cybersecurity Analysis

- General Workflow: ETL $\rightsquigarrow$ ‘Analytics’ $\rightsquigarrow$ ‘Outliers’ $\rightsquigarrow$ ‘Deep Dive’
- Push Button Time Allocation within Data Scientists:
    - Analytics ($\sim$ 25%)
    - Deep Dive ($\sim$ 75%)

A ~~Deep Dive~~ Dip Into Deep Dive

- Manual sifting of small data
- Find ‘interesting’ outliers
- Tell story about those outliers
- Including, why the outlier was flagged as such

Aim: Better ways of learning about, and from outliers.

# Better ways of learning about, and from outliers

# Better ways of learning about, and from outliers

- Three components:

# Better ways of learning about, and from outliers

- Three components:
    - Data driven account of why an outlier is an outlier

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly
- (Better) Know Your IP

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly
- (Better) Know Your IP
  - Gets ‘known knowns’ and ‘easily knowns’

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly
- (Better) Know Your IP
  - Gets 'known knowns' and 'easily knowns'
  - Where: Lat/Long, City, Country, Time zone

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly
- (Better) Know Your IP
  - Gets 'known knowns' and 'easily knowns'
  - Where: Lat/Long, City, Country, Time zone
  - Which ports are open, which services are running

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly
- (Better) Know Your IP
  - Gets ‘known knowns’ and ‘easily knowns’
  - Where: Lat/Long, City, Country, Time zone
  - Which ports are open, which services are running
  - Ping and traceroute

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly
- (Better) Know Your IP
  - Gets 'known knowns' and 'easily knowns'
  - Where: Lat/Long, City, Country, Time zone
  - Which ports are open, which services are running
  - Ping and traceroute
  - Blacklisted or not, and for what

# Better ways of learning about, and from outliers

- Three components:
  - Data driven account of why an outlier is an outlier
  - Data driven inference about patterns of outliers
  - Diving Deeply More Quickly
- (Better) Know Your IP
  - Gets ‘known knowns’ and ‘easily knowns’
  - Where: Lat/Long, City, Country, Time zone
  - Which ports are open, which services are running
  - Ping and traceroute
  - Blacklisted or not, and for what
  - Zmap and Zgrab scans of IPv4

# Notes and Concerns

- Geocoding:

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.
  - Average geolocation error can be hefty. For instance, GeoIPInfo puts me $\sim$ 30 miles away

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.
  - Average geolocation error can be hefty. For instance, GeoIPInfo puts me $\sim$ 30 miles away
- Time Zone

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.
  - Average geolocation error can be hefty. For instance, GeoIPInfo puts me $\sim$ 30 miles away
- Time Zone
  - Globe split into 24 Time Zones. Conditional on lat/long, we know time zone.

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.
  - Average geolocation error can be hefty. For instance, GeoIPInfo puts me $\sim$ 30 miles away
- Time Zone
  - Globe split into 24 Time Zones. Conditional on lat/long, we know time zone.
  - But countries create own rules

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.
  - Average geolocation error can be hefty. For instance, GeoIPInfo puts me $\sim$ 30 miles away
- Time Zone
  - Globe split into 24 Time Zones. Conditional on lat/long, we know time zone.
  - But countries create own rules
  - India, for instance, has 30 minute offset. Or for e.g., Mountain Time

## Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.
  - Average geolocation error can be hefty. For instance, GeoIPInfo puts me $\sim$ 30 miles away
- Time Zone
  - Globe split into 24 Time Zones. Conditional on lat/long, we know time zone.
  - But countries create own rules
  - India, for instance, has 30 minute offset. Or for e.g., Mountain Time
- Active scanning (ping, traceroute)

# Notes and Concerns

- Geocoding:
  - Start with DB of known 'landmarks'
  - Compute maximum distance to last router from multiple landmarks using max. Internet speed
  - Gives an approximate bounding box. Take centroid of that.
  - Average geolocation error can be hefty. For instance, GeoIPInfo puts me $\sim$ 30 miles away
- Time Zone
  - Globe split into 24 Time Zones. Conditional on lat/long, we know time zone.
  - But countries create own rules
  - India, for instance, has 30 minute offset. Or for e.g., Mountain Time
- Active scanning (ping, traceroute)
  - You give away your location. `tcpdump`.

## Know, Know Your IP

# Know, Know Your IP

- Links to:
  - GeoNames
  - MaxMind
  - Virustotal
  - IpVoid
  - AbuseIPDB
  - Shodan
  - Censys
  - Ping
  - Traceroute
  - tzwhere

# Know, Know Your IP

- Links to:
  - GeoNames
  - MaxMind
  - Virustotal
  - IpVoid
  - AbuseIPDB
  - Shodan
  - Censys
  - Ping
  - Traceroute
  - tzwhere
- Rate limits for free tier

## Know, Know Your IP

- Links to:
  - GeoNames
  - MaxMind
  - Virustotal
  - IpVoid
  - AbuseIPDB
  - Shodan
  - Censys
  - Ping
  - Traceroute
  - tzwhere
- Rate limits for free tier
- Use it for a handful of IPs

# Know, Know Your IP

# Know, Know Your IP

- Installation

```
pip install know_your_ip

# If traceroute not installed on Linux
sudo apt-get install traceroute
```

# Know, Know Your IP

- Installation

```
pip install know_your_ip

# If traceroute not installed on Linux
sudo apt-get install traceroute
```

- Components:

# Know, Know Your IP

- Installation

```
pip install know_your_ip

# If traceroute not installed on Linux
sudo apt-get install traceroute
```

- Components:

1. Configuration file:

```
know_your_ip.cfg

[abuseipdb]
enable = 1
user_id = 1234
key = a0fbe08ccef4924517949071 3e551b589
cat_catid = abuseipdb_cat_catid.csv

[ipvoid]
enable = 1
```

# Components of KIP

2. What columns do you want

```
columns.txt

# Ping
ping.timeout
ping.count
ping.max
 ...

# abuseipdb API
abuseipdb.bad_isp
abuseipdb.categories
abuseipdb.reports
abuseipdb.total
 ...
```

# Using KIP

```
python know_your_ip.py

# Paths
--file path_to_input_file (default input.csv)
--config path_config_file   (default know_your_ip.cfg)
--output path_to_output_file (default output.csv)

# Max connections (multi-threaded)
--maxconn MAX_CONN

# From/to Row
--from from_row
--to to_row

# Verbose
--verbose verbose
```

# Actually Using KIP

```
# For one/few IP(s)
python know_your_ip.py 94.31.29.154
python know_your_ip.py 94.31.29.154 204.2.197.211

# File, some rows
python know_your_ip.py --file input_small.csv --from 1
 --to 2
```