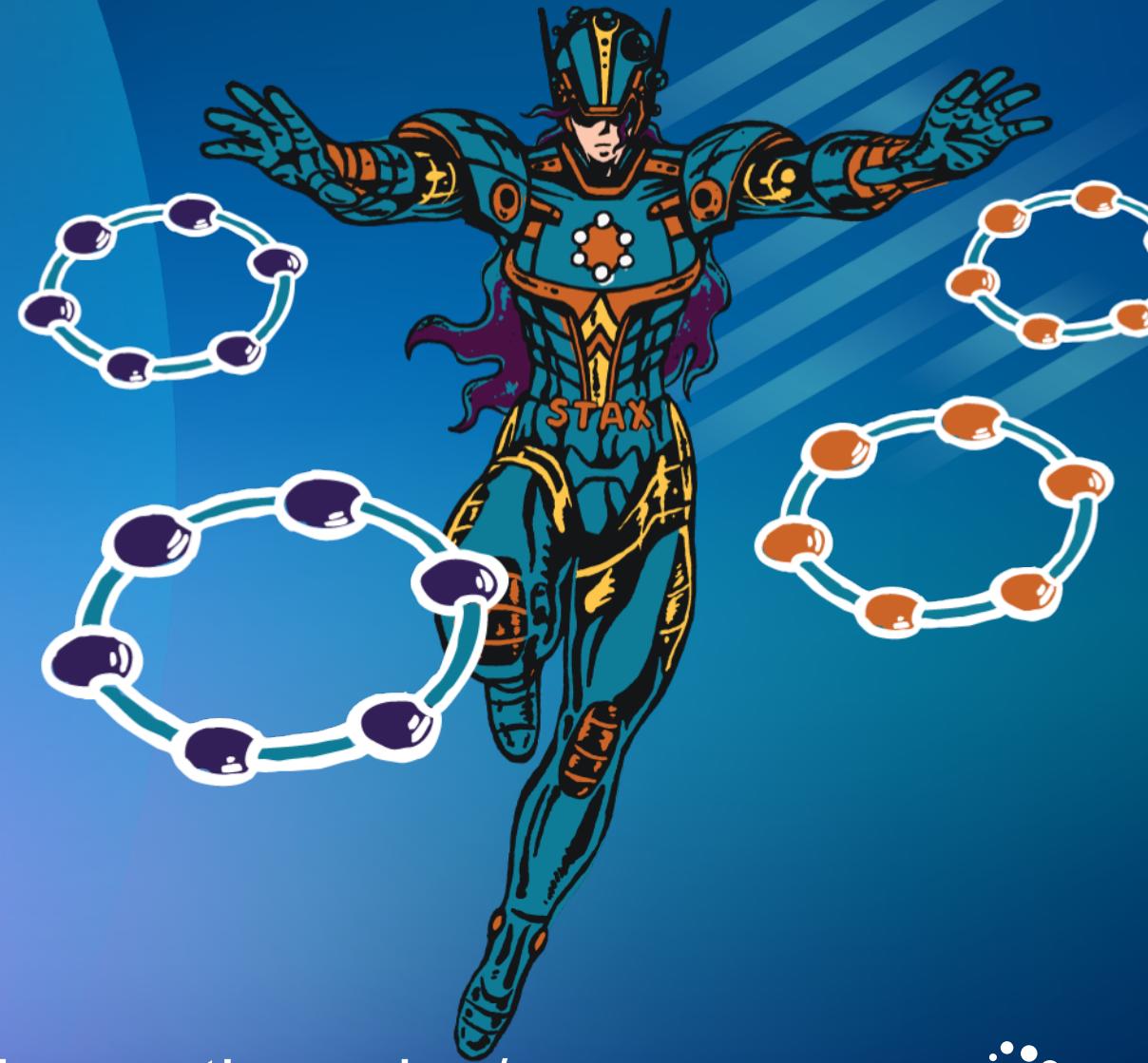


# DATASTAX **ACCELERATE**

Predicting Restaurant Inspection Failures



<https://github.com/angelok1/chicago-inspections-dse/>

DATASTAX®

# About Us

## Me

- Help organizations implement Health IT (FHIR), Data Science (ML, DNN), Big Data (Spark, DSE, Kafka)
- Graduate researcher at Harvard Data Systems Laboratory



[github.com/angelok1](https://github.com/angelok1)



<https://www.linkedin.com/in/angelok1/>



[akastroulis@carrera.io](mailto:akastroulis@carrera.io)



[@carraio](https://twitter.com/carraio)

## You

- Data Scientists (build Models)?
- Engineers (implement Models)?
- New to Data Science (Just want to know more about what this thing is and how it works)?

# The Project



<https://github.com/angelok1/chicago-inspections-dse>



**CHICAGO  
DATA PORTAL**

<https://data.cityofchicago.org/>

# The Toolbox



The screenshot shows three Jupyter Notebook windows side-by-side. The left window is the 'Welcome to P...' page. The middle window is an 'Exploring the Lorenz System' notebook cell containing code and sliders for parameters. The right window is a plot of the Lorenz attractor.

**Jupyter Welcome to P...**

**In [ ]:**

```
matplotlib inline
import pandas as pd
import numpy as np
import matplotlib
```

**jupyter Exploring the Lorenz System**

In this Notebook we explore the [Lorenz system](#) of differential equations:

$$\dot{x} = \sigma(y - x)$$
$$\dot{y} = \rho x - y - xz$$
$$\dot{z} = -\beta z + xy$$

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters ( $\sigma$ ,  $\beta$ ,  $\rho$ ) are varied, including what are known as *chaotic solutions*. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

**In [7]:**

```
interact(Lorenz, N=fixed(10), angle=(0.,360.),
          σ=(0.0,50.0),β=(0.,5), ρ=(0.0,50.0))
```

angle: 308.2

max\_time: 12

σ: 10

β: 2.6

ρ: 28

4

# The Problem

- According to the CDC, 47.8M cases of foodborne illness annually
- Leading to 3,000 deaths
- 31 known pathogens but 38.4M come from unspecified agents
- Norovirus, Salmonella, Staph, E. Coli and a bunch of stuff we have no idea what they are.
- Seems like an ideal ML problem

# Building Intuition

If we could discover food borne illness quicker, we'd have less exposure.

Where do we get our food?

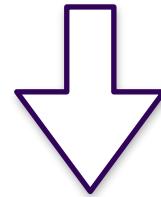
Data?



# Building Intuition

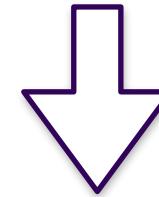
If we could discover food borne illness quicker, we'd have less exposure.

Where do we get our food?



Restaurants, Grocery

Data?



Inspections, Business Licenses



# Building Intuition



# Building Intuition - QUIZ

**Which of these pairs do you think will be most relevant?**



Complaint of food poisoning?

Did it snow?

Number of burglaries in the area?

Do they serve sushi?

Garbage in the alley?

How often they are inspected?

Close to a Walmart Supercenter?

Close to the water?

# Building Intuition - QUIZ

**Which of these pairs do you think will be most relevant?**

Complaint of food poisoning?

**Did it snow?**

**Number of burglaries in the area?**

Do they serve sushi?

Garbage in the alley?

**How often they are inspected?**

**Close to a Walmart Supercenter?**

Close to the water?

# Building Intuition

“All models are wrong, but some are useful”

— George Box



# Building Intuition

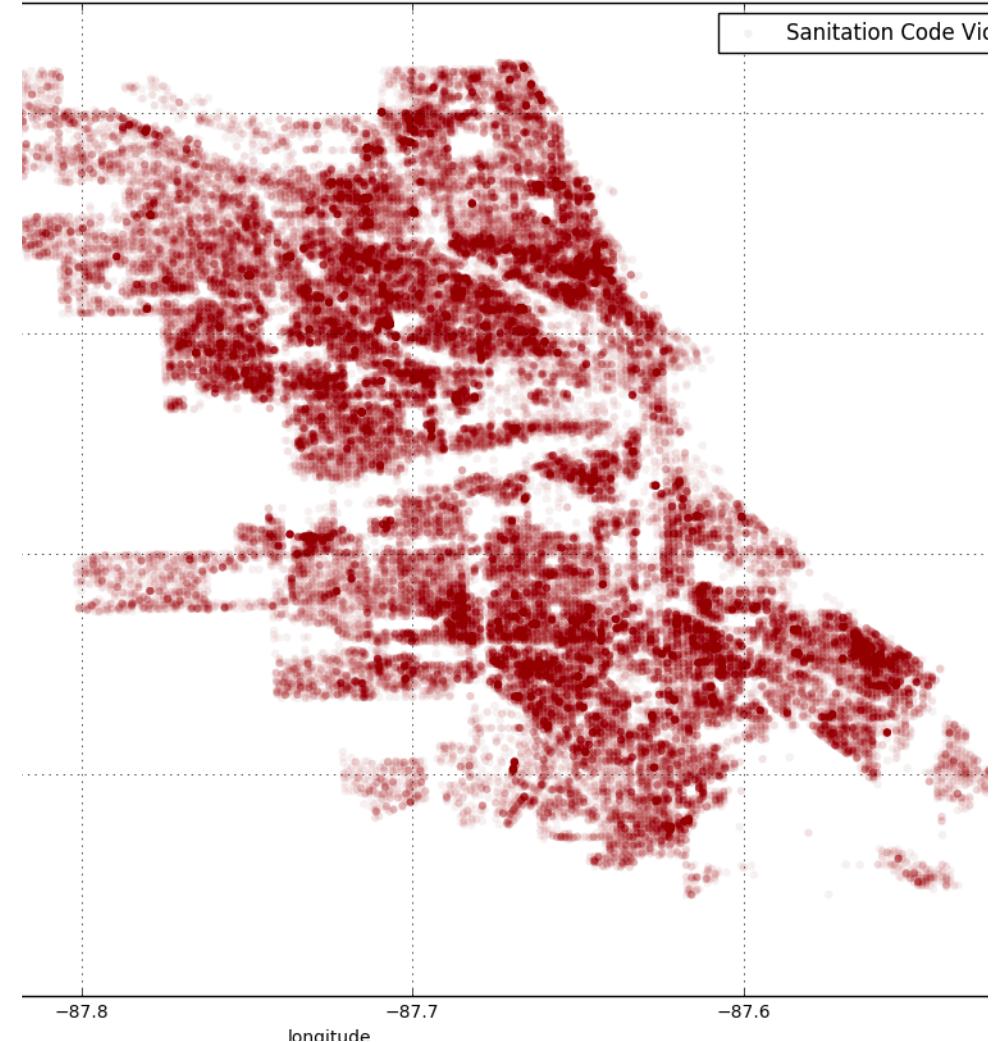
- Do **sanitation** complaints have anything to do with failures?
  - Where are rodents or bugs likely to infest?
  - If the alley is full of trash, how clean is the neighborhood?
- Does **crime** have anything to do with failures?
  - What if it was a burglary and affected the power or equipment?
- How does **weather** have an effect on equipment?



# Exploring Sanitation

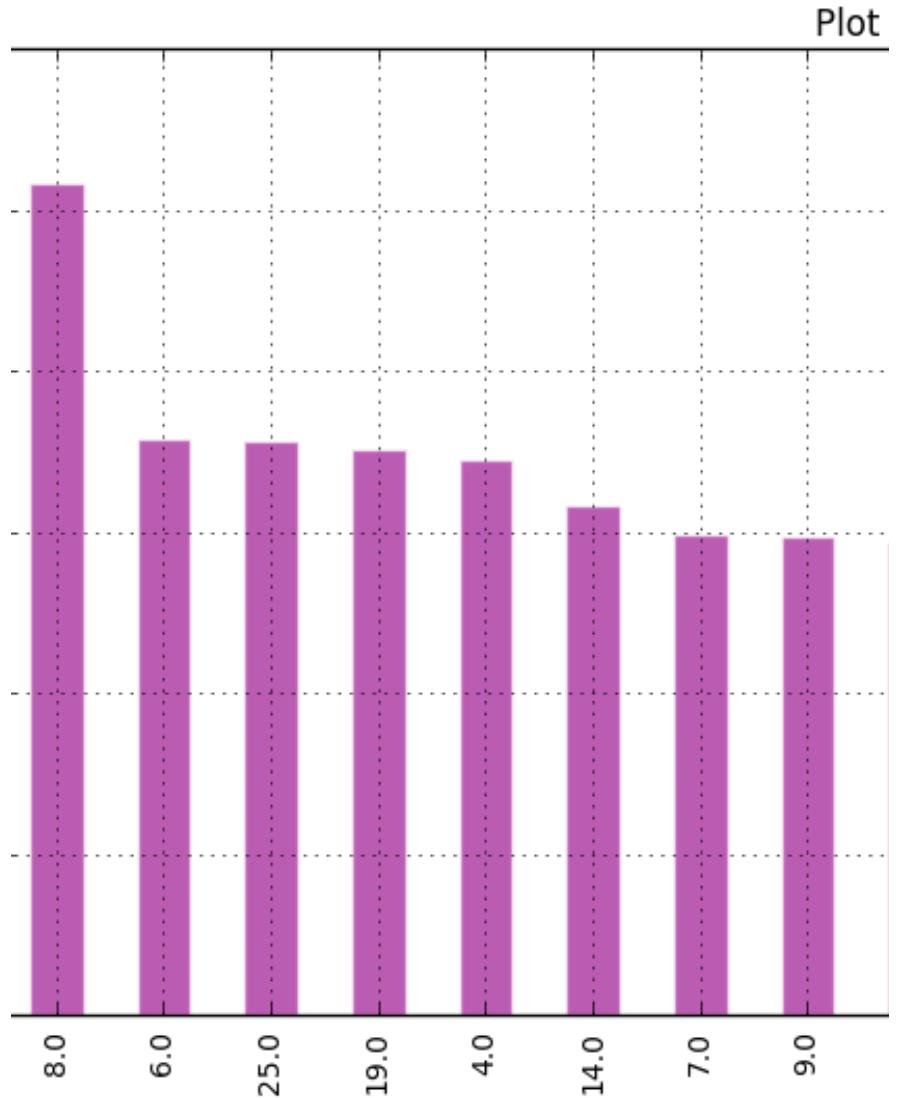
```
scatter_plot(x='longitude', y='latitude', my_df=df_sanitation,  
group='type_of_service_request',  
col=[ "darkred"], alpha=0.05, size=20, marker='o',  
title="Plot of the sanitation code violations in Chicago")
```

Plot VIII: Plot of the sanitation code violations in Chicago



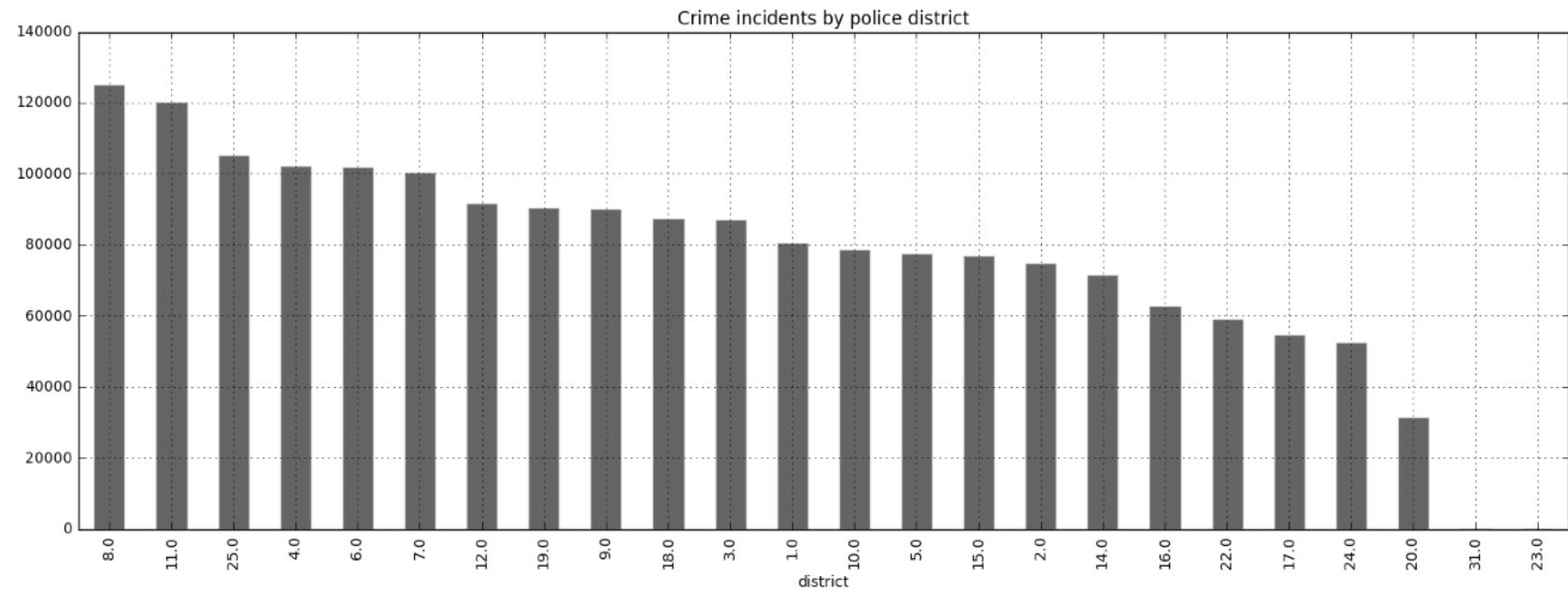
# Exploring Sanitation

```
bar_plot(my_df=df_sanitation, var_name='police_district',  
        color='purple', alpha=0.6,  
        title='Number of sanitation code complaints by police district',  
        edgecolor='white', figsize=(18, 6))
```



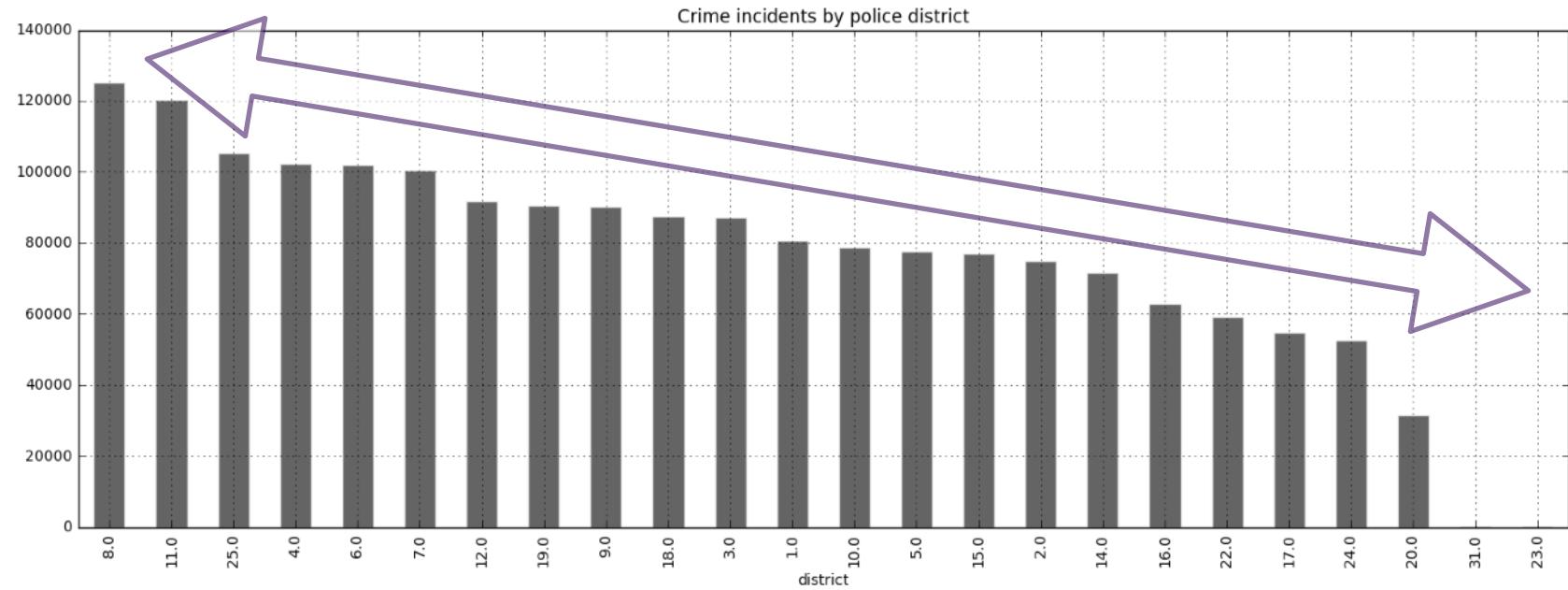
# Exploring Crime

```
bar_plot(my_df=df_crimes, var_name='district', color='black', alpha=0.6,  
        title='Crime incidents by police district', edgecolor='white', figsize=(18, 6))
```



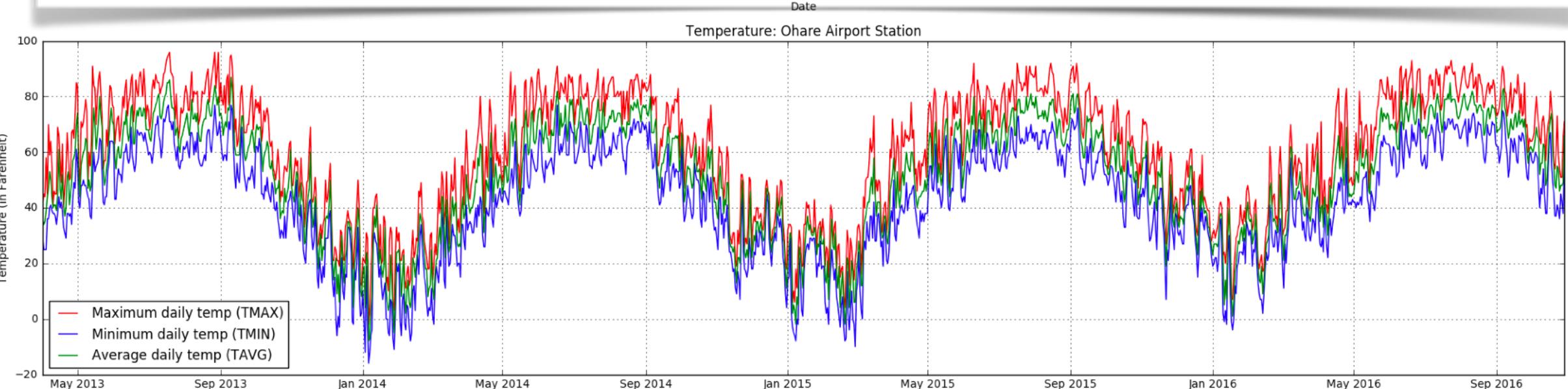
# Exploring Crime

```
bar_plot(my_df=df_crimes, var_name='district', color='black', alpha=0.6,  
        title='Crime incidents by police district', edgecolor='white', figsize=(18, 6))
```



# Exploring Weather

```
ax2.plot(df_ohareairport[ 'DATE' ], df_ohareairport[ 'TMAX' ], c='r', label='Maximum daily temp (TMAX)')  
ax2.plot(df_ohareairport[ 'DATE' ], df_ohareairport[ 'TMIN' ], c='b', label='Minimum daily temp (TMIN)')  
ax2.plot(df_ohareairport[ 'DATE' ], df_ohareairport[ 'TAVG' ], c='g', label='Average daily temp (TAVG)')  
  
ax2.set_title('Temperature: Ohare Airport Station')  
ax2.set_xlabel('Date')  
ax2.set_ylabel('Temperature (in Farenheit)')  
ax2.legend(loc='best')  
ax2.grid()
```



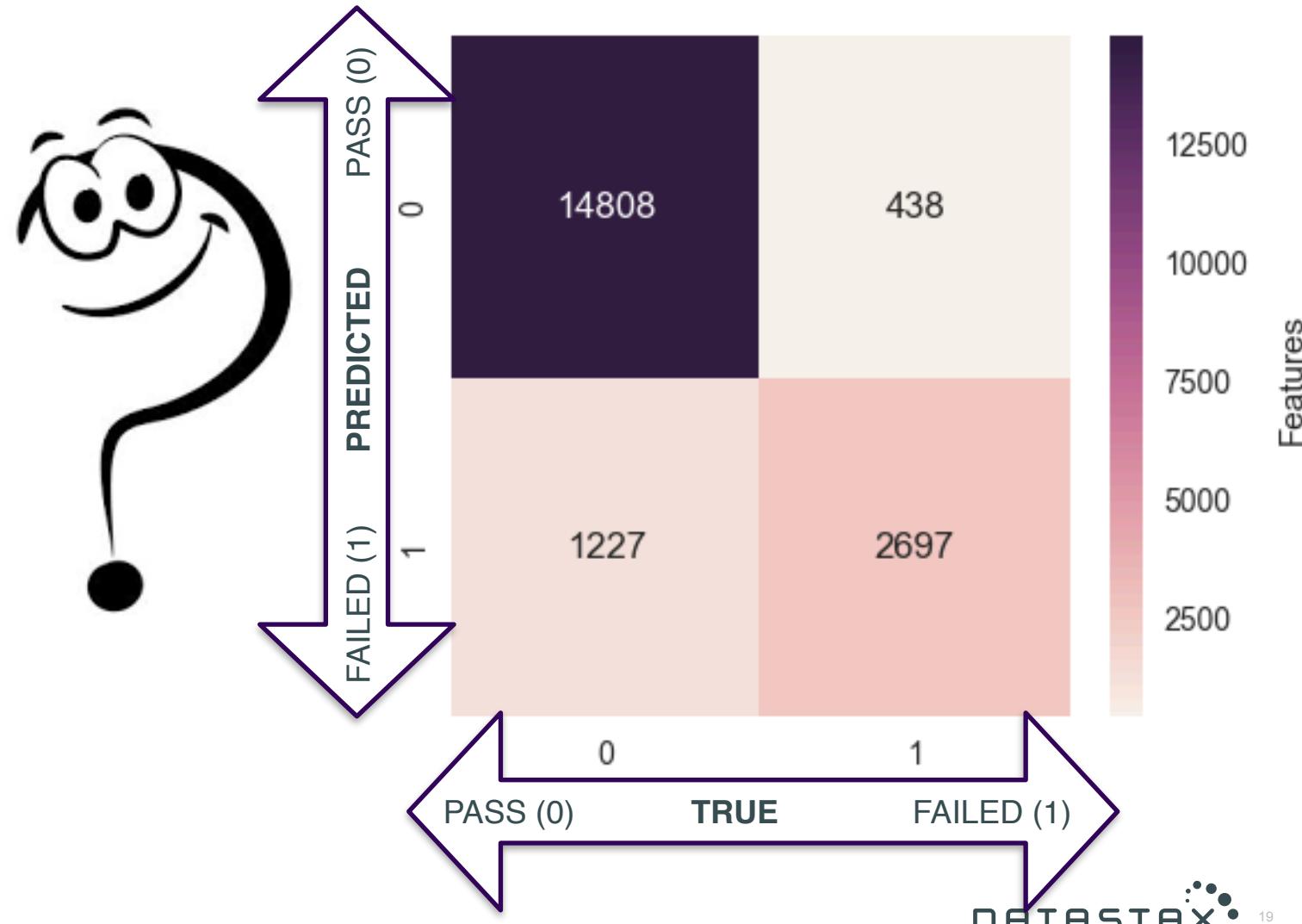
# Modeling

- What are we trying to **accomplish**?
  - We want something to sort on.
- How do we **model** this?
  - NN? time series? ML?
- What kinds of **decisions** do we make?
  - How engineer features? Preprocess?
  - Where to store data (DSEFS vs C\*)?
  - How will we run this model (Real time vs Batch)?



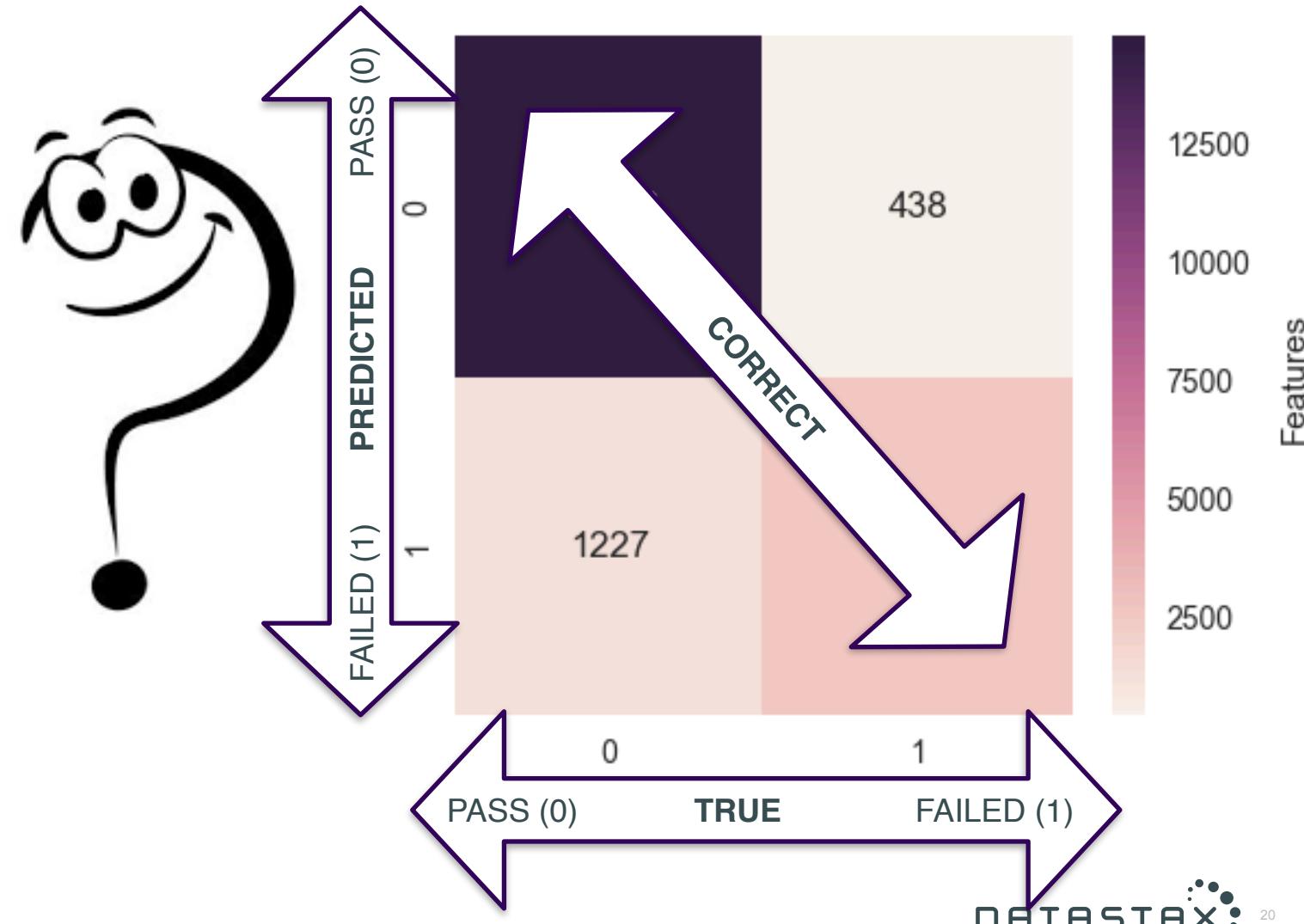
# Model accuracy

- Do we use accuracy as the measure of quality (i.e. actual - predicted)
- Remember **everyone** gets inspected.



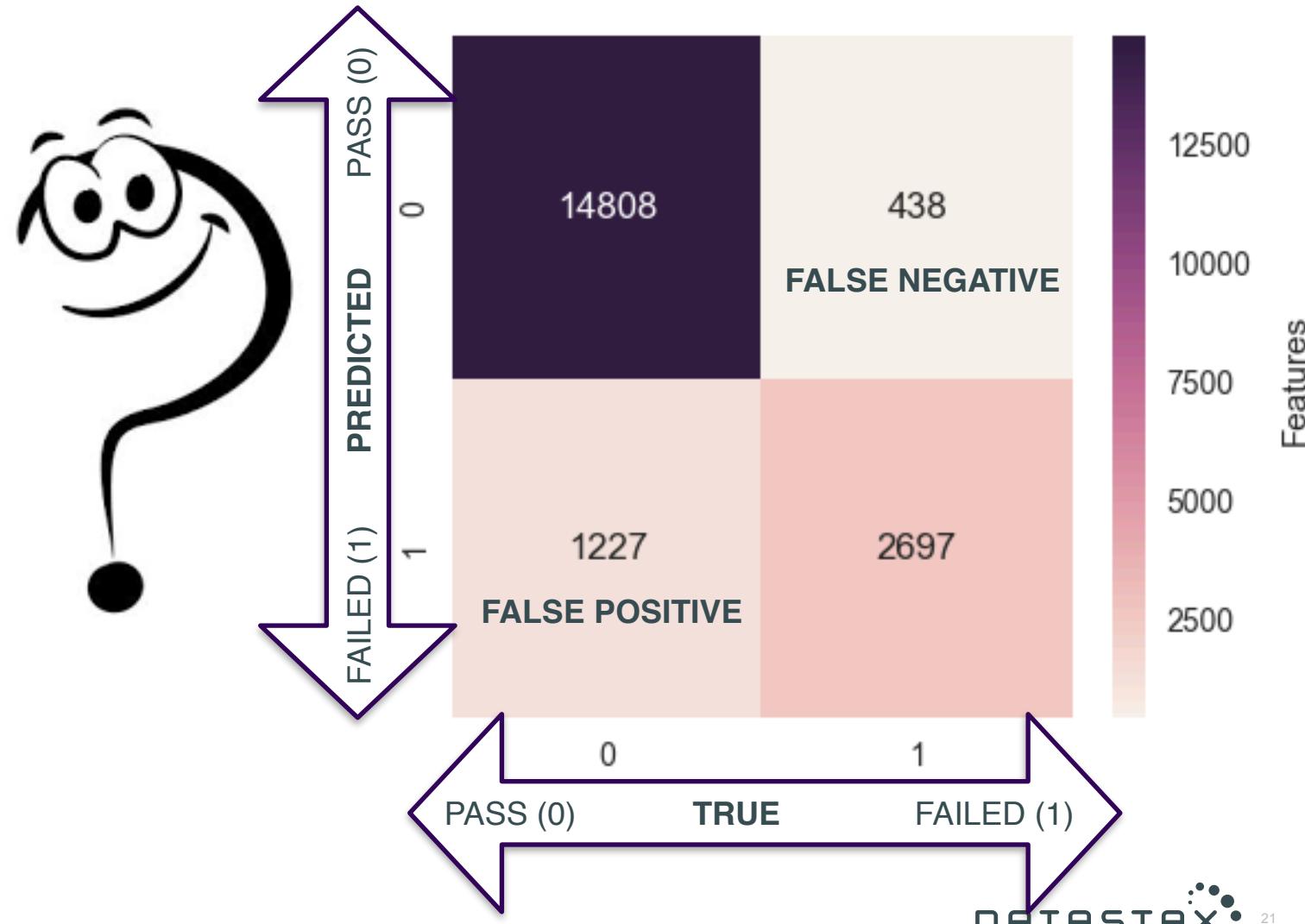
# Model accuracy

- Do we use accuracy as the measure of quality (i.e. actual - predicted)
- Remember **everyone** gets inspected.

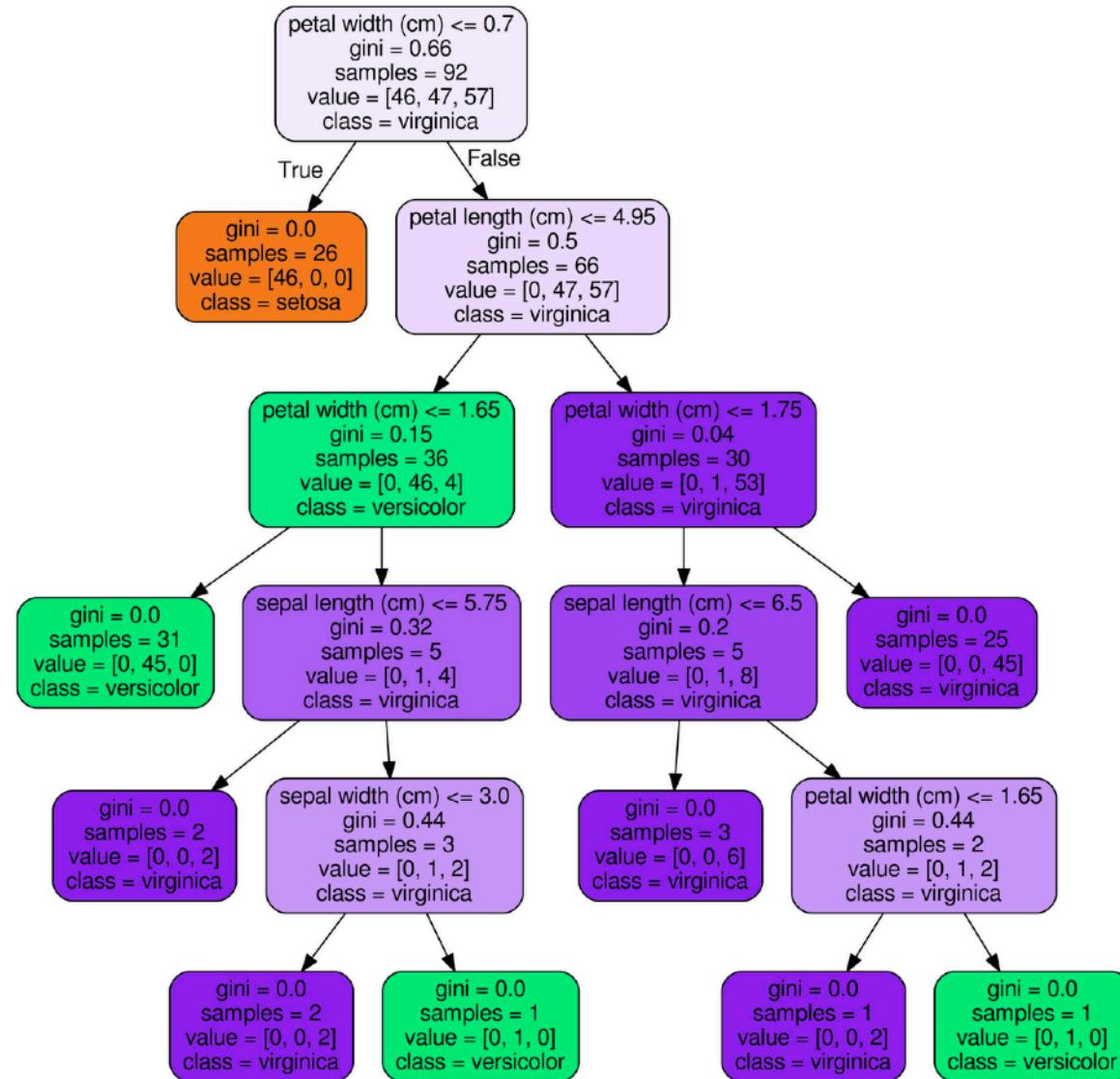


# Model accuracy

- Do we use accuracy as the measure of quality (i.e. actual - predicted)
- Remember **everyone** gets inspected.

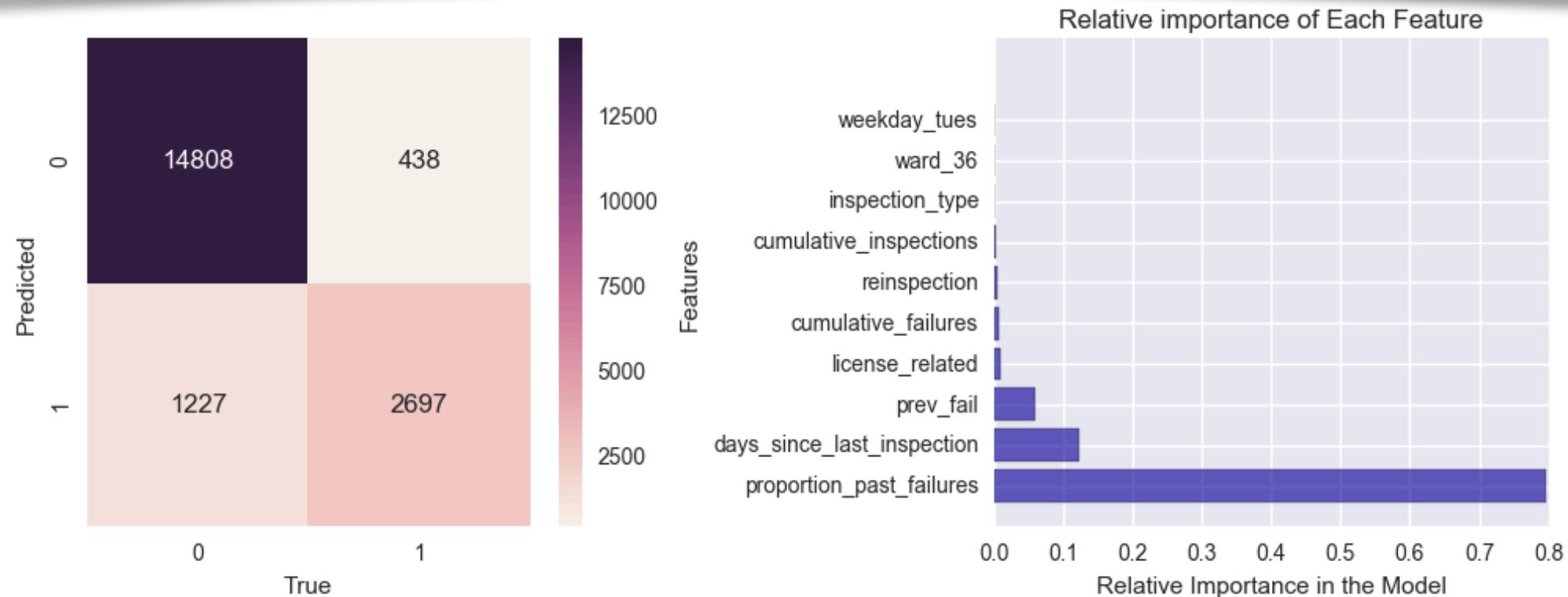


# Decision Tree Classification



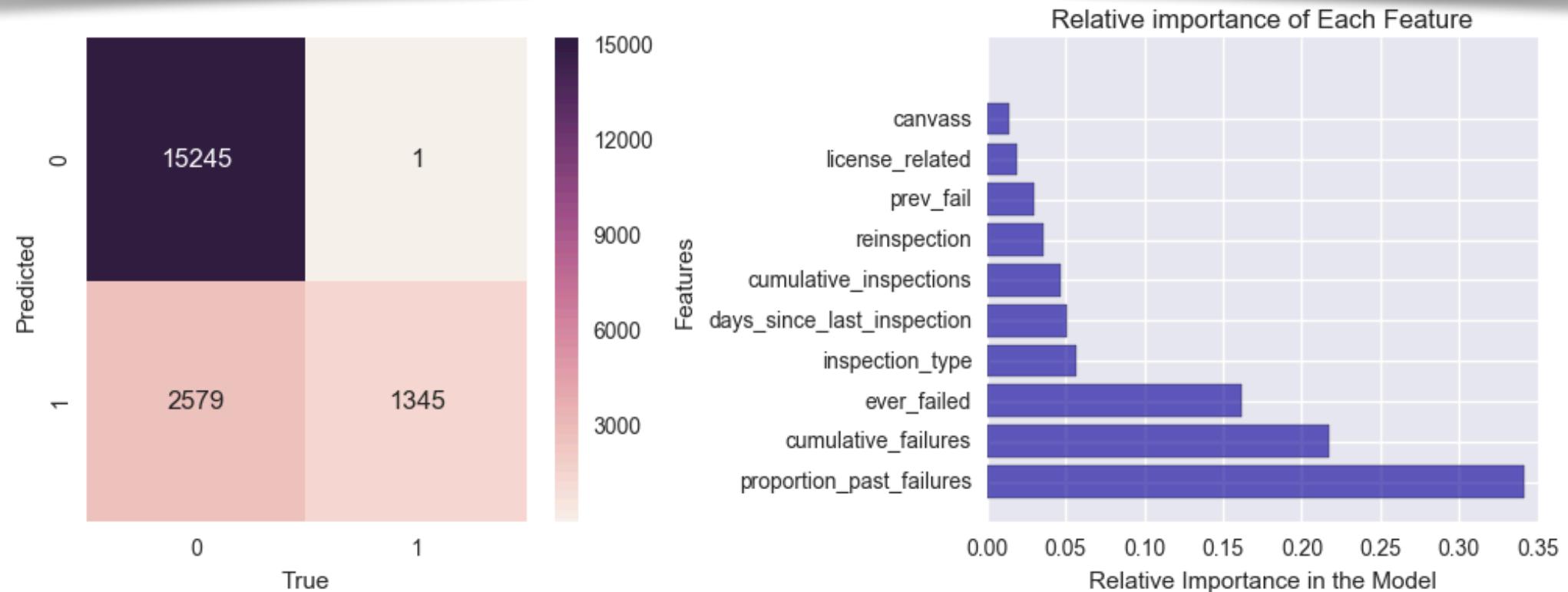
# Modeling - Decision Trees

```
dt = DecisionTreeClassifier(labelCol="label", featuresCol="features")
model = dt.fit(train)
predictions = model.transform(test)
reportResults(predictions.select("prediction", "label").rdd, model)
```



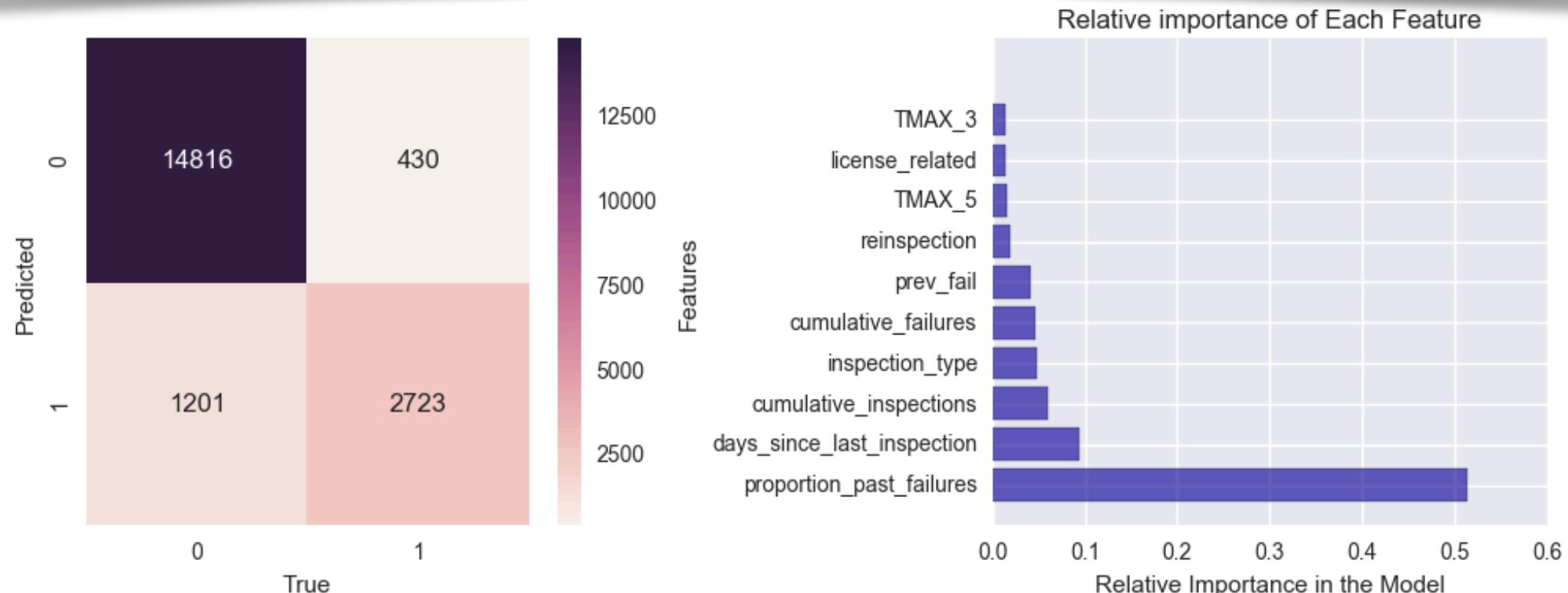
# Modeling - Random Forest

```
rf = RandomForestClassifier(labelCol="label", featuresCol="features", numTrees=500)
model = rf.fit(train)
predictions = model.transform(test)
reportResults(predictions.select("prediction", "label").rdd, model)
```

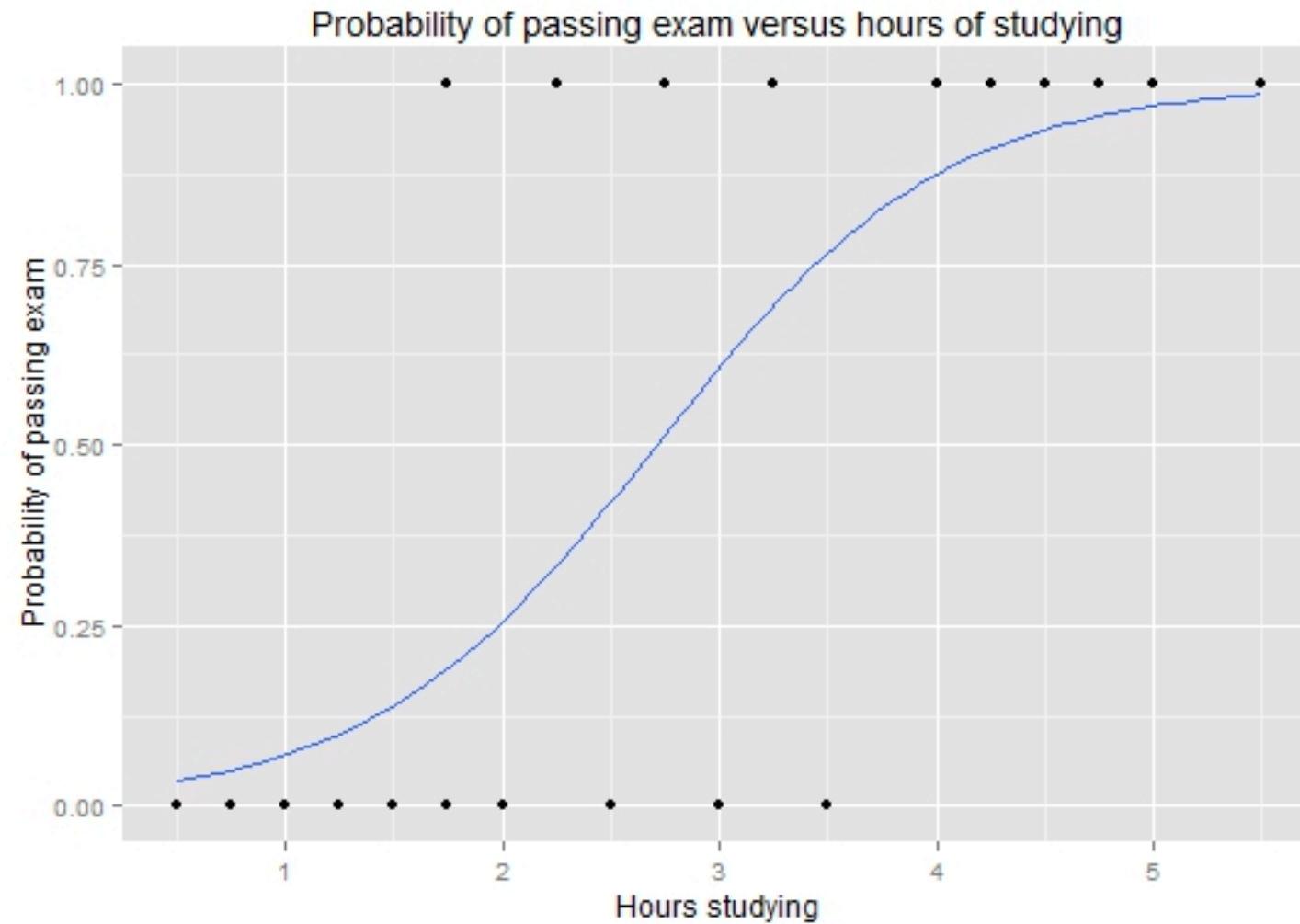


# Modeling - Gradient Boosted Trees

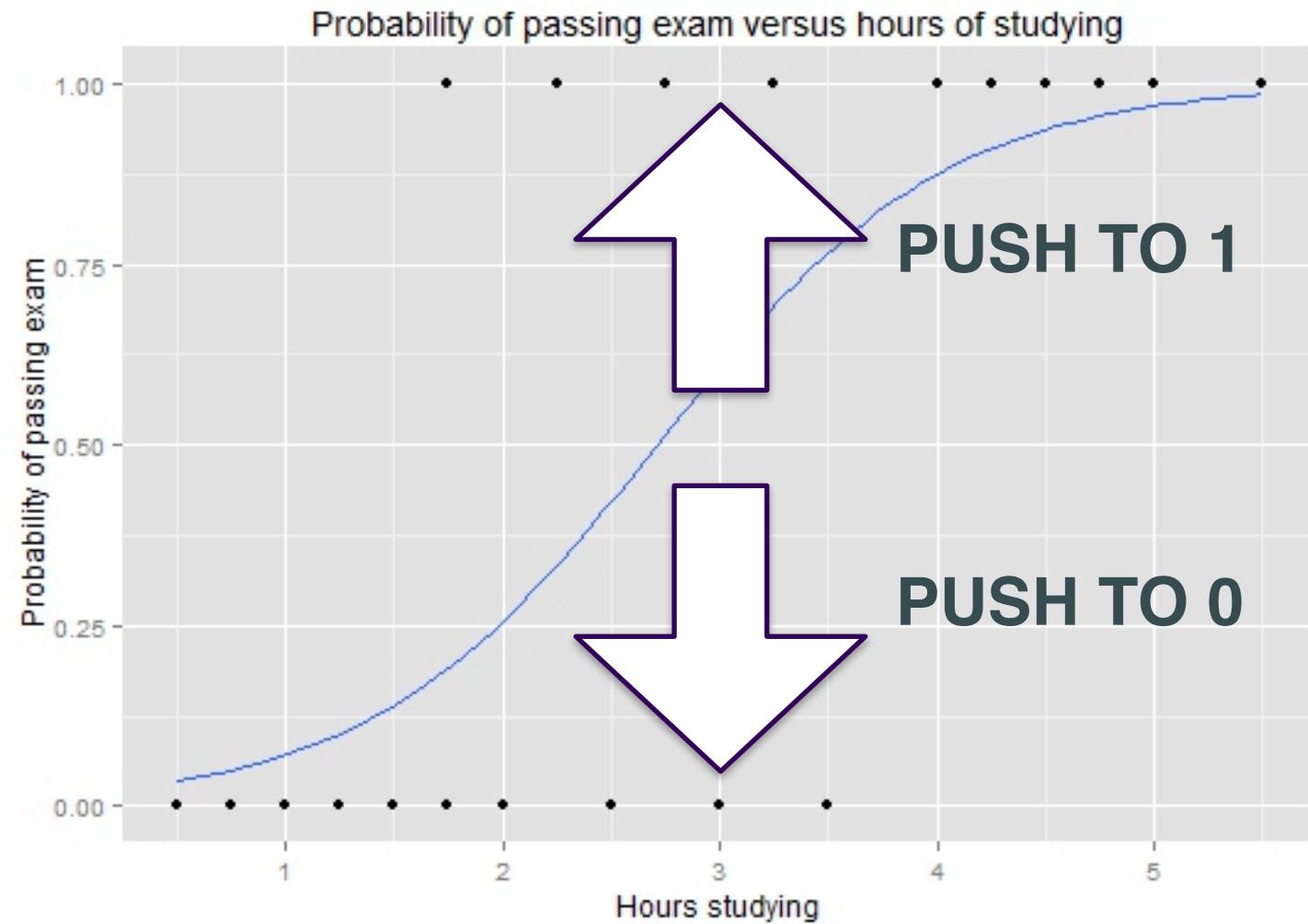
```
gbt = GBTClassifier(labelCol="label", featuresCol="features", maxIter=100)
gbt_model = gbt.fit(train)
predictions = gbt_model.transform(test)
reportResults(predictions.select("prediction", "label").rdd, gbt_model)
```



# Modeling - Weighted Logistic Regression



# Modeling - Weighted Logistic Regression



# Modeling - Weighted Logistic Regression

```
weighted_train = train.withColumn("logistic_weights",
    when(col("label")==0.0, 0.25).otherwise(0.75))

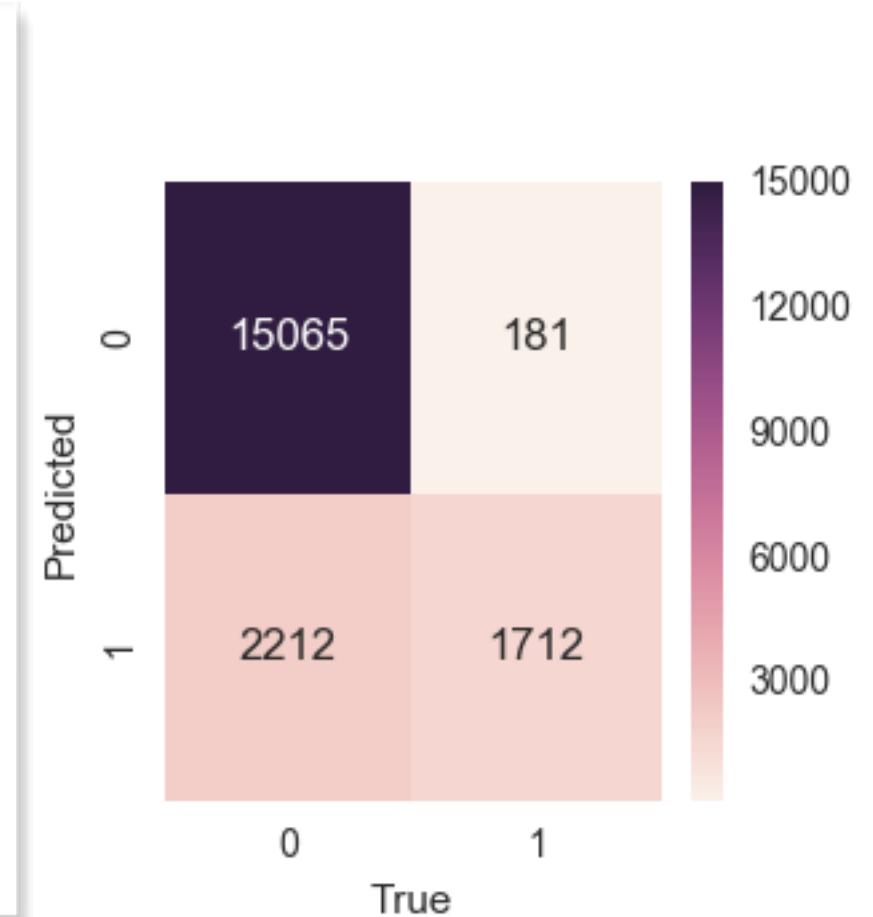
lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8,
    weightCol="logistic_weights")

model = lr.fit(train)

predictions = model.transform(test)

metrics = MulticlassMetrics(predictions.select("prediction", "label").rdd)

reportAccuracy(metrics)
```



# Production

- How do we get this to production?
- What if there's not parity between tools?
  - i.e. Scikit vs SparkML, SparkDF vs Pandas
- Where does the data live?
  - DSEFS? C\*?



# ONNX



- An open format to represent deep learning models

Supports

LibSVM

APACHE  
Spark™

TensorFlow

Keras

dmlc  
**XGBoost**

scikit  
learn

Supported By

Facebook  
Open Source

aws

Microsoft

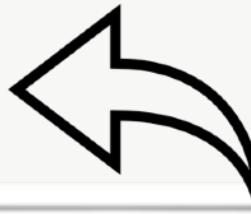
# ONNX



```
pipeline_model = pipeline.fit(training_data)

onnx_model = convert_sparkml(pipeline_model, 'My Sparkml Pipeline', initial_types)

with open("model.onnx", "wb") as f:
    f.write(onnx_model.SerializeToString())
```



## SparkML

## ONNX Framework

```
onnx_model = onnx.load('path/to/the/model.onnx')

sess = onnxruntime.InferenceSession(onnx_model)

output = sess.run(None, input_data)
```



# Koalas

```
import pandas as pd
import databricks.koalas as ks

pdf = pd.DataFrame({'x':range(3), 'y':['a','b','b'],
'z':['a','b','b']})

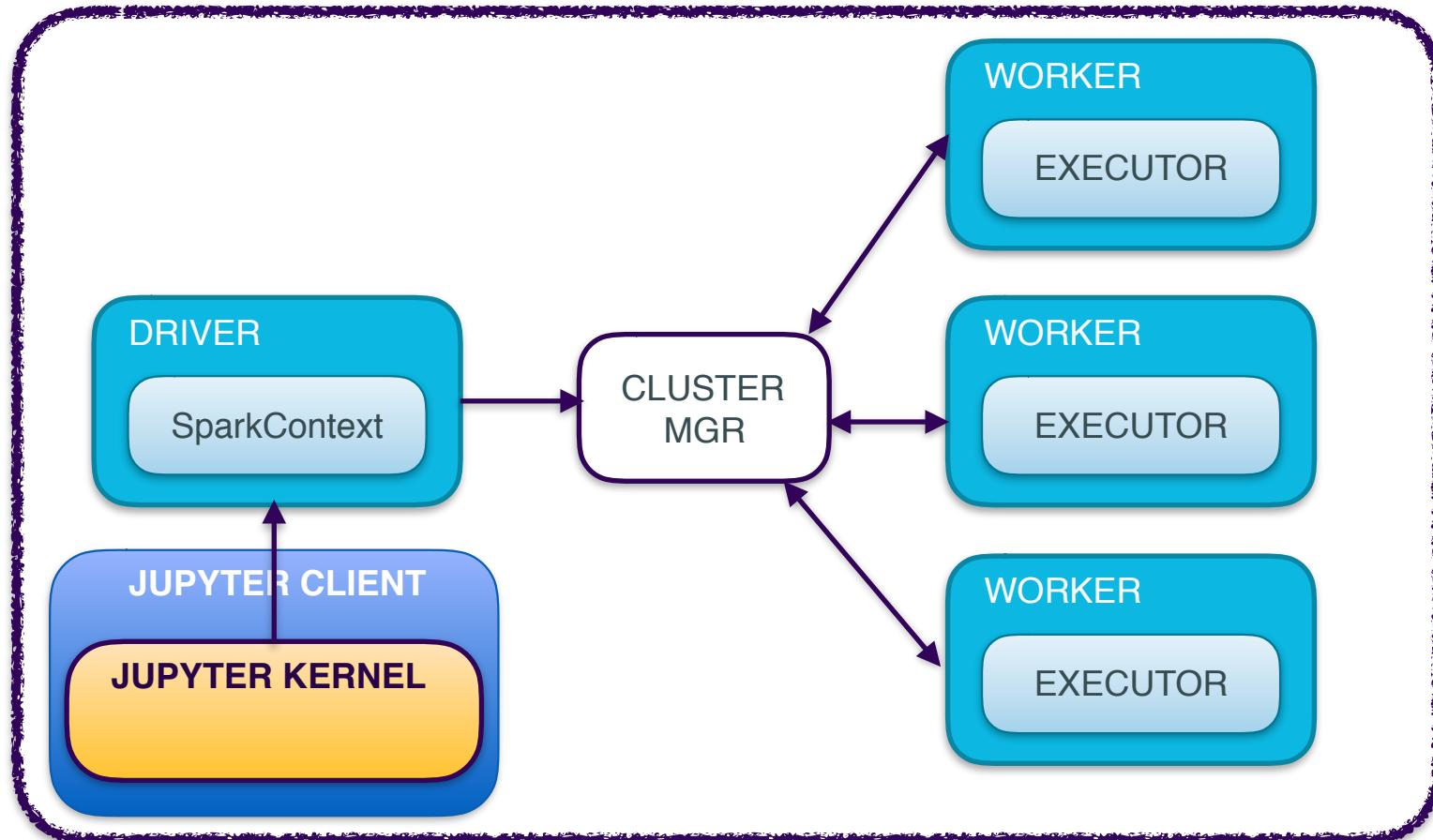
# Create a Koalas DataFrame from pandas DataFrame
df = ks.from_pandas(pdf)

# Rename the columns
df.columns = ['x', 'y', 'z1']

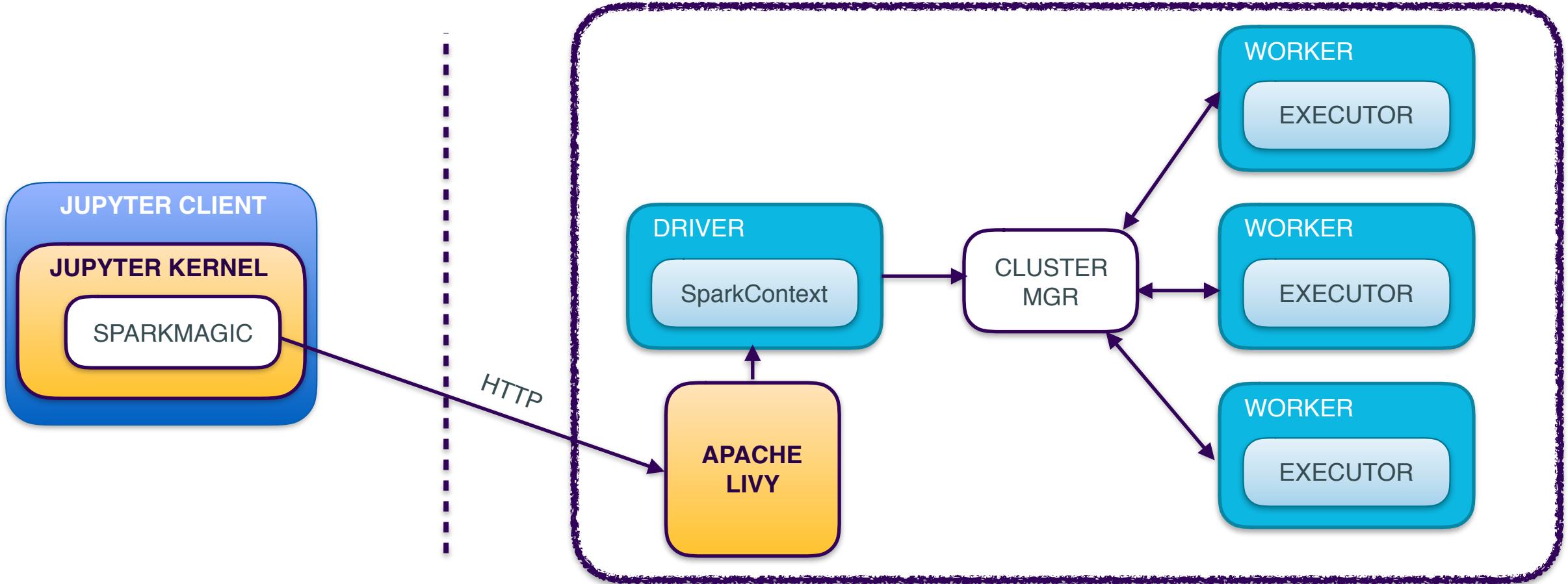
# Do some operations in place:
df['x2'] = df.x * df.x
```

- Parity with pandas!
- It's New (VERY new)
- Support Spark 2.4, not tested on older versions

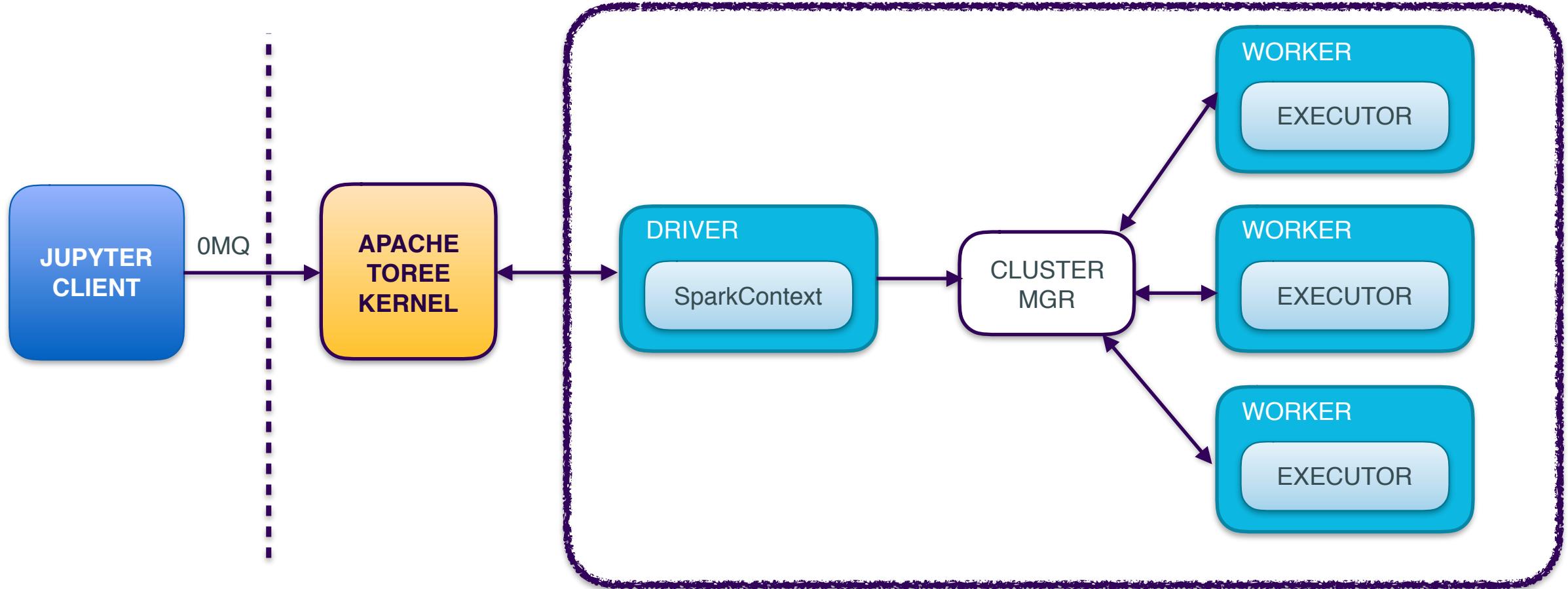
# Jupyter with Spark



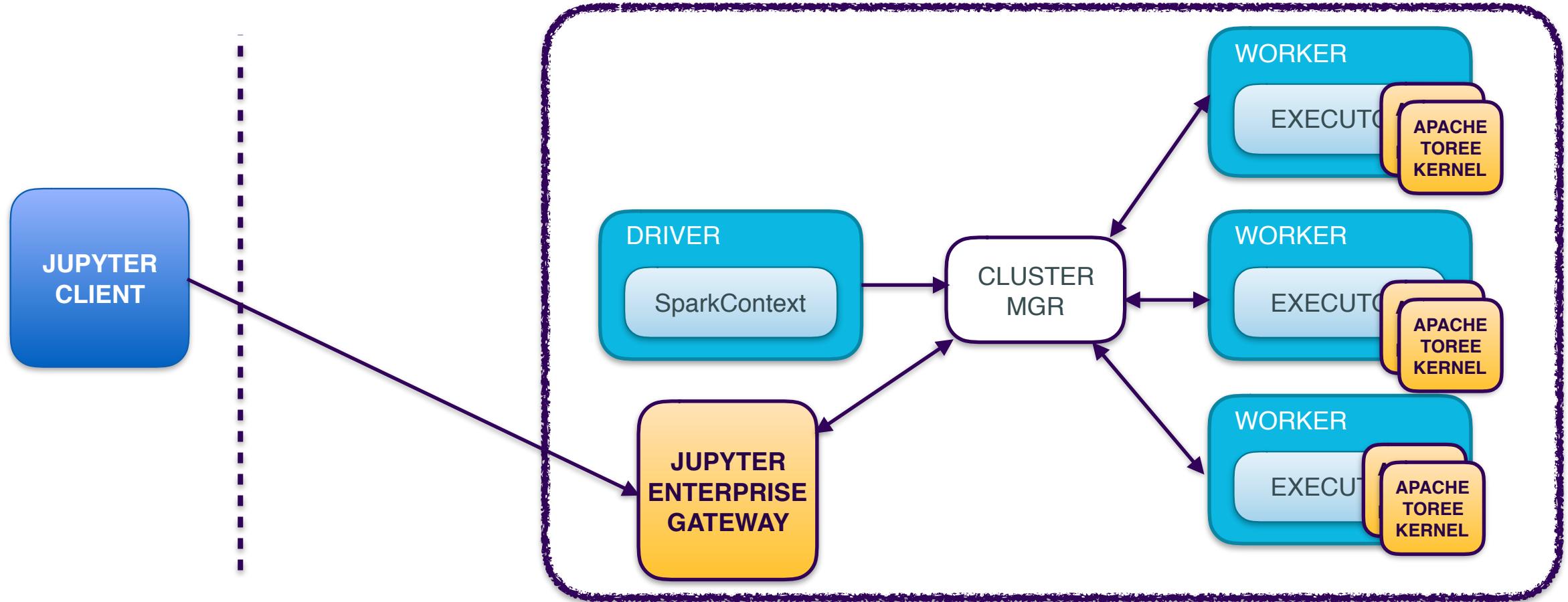
# Jupyter with Spark (SparkMagic & Livy)



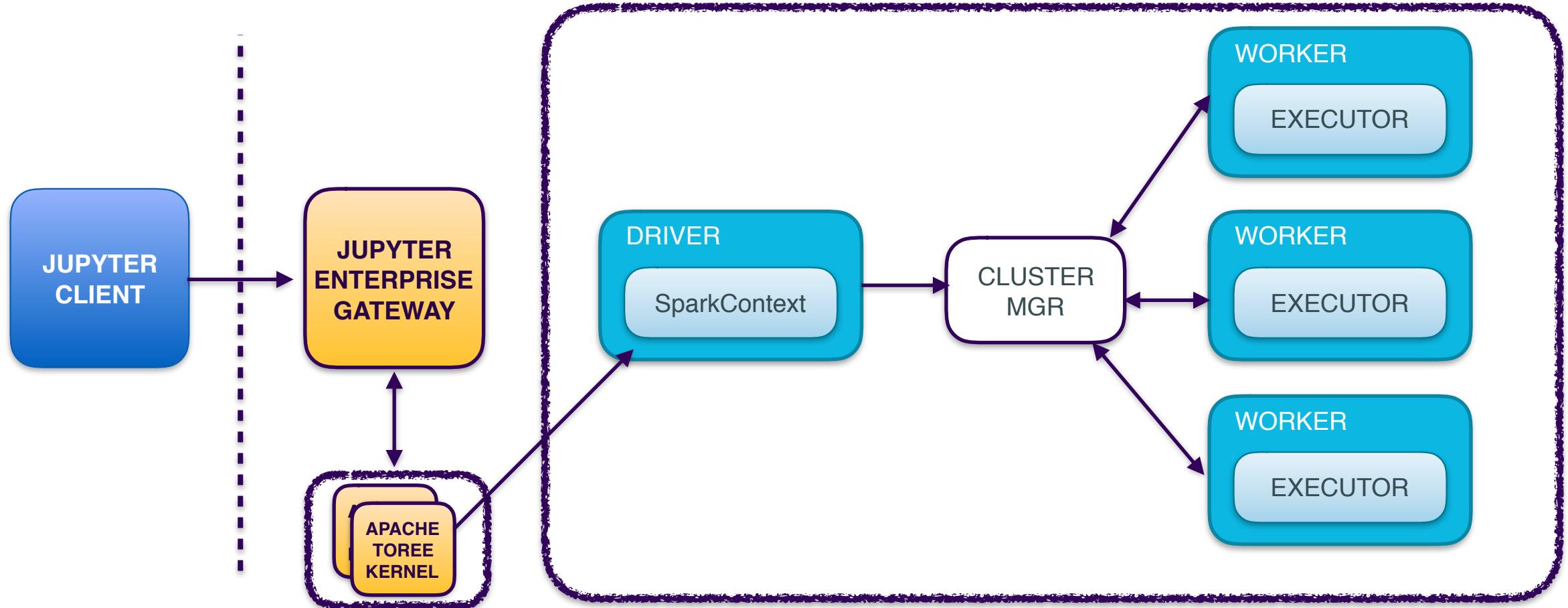
# Jupyter with Spark (Apache Toree)



# Jupyter with Spark (Jupyter Enterprise Gateway)



# Jupyter with Spark (Jupyter Enterprise Gateway)



# Final Thoughts

- Challenge your intuition
- The models we make do not live in a vacuum
- Don't make everyone use the same tools for everything. The platform should be flexible



DATASTAX  
**ACCELERATE**  
THANK YOU

<https://github.com/angelok1/chicago-inspections-dse/>  
<https://www.linkedin.com/in/angelok1/>  
akastroulis@carrera.io