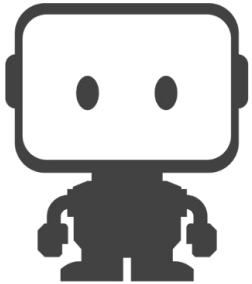
ML for Developers

Commoditization of ML



DataRobot

ML Workflow

1. Building original model
2. Deploying a trained model

Building an Original Model

Get the data:

How much data do we need?

MORE.

Building an Original Model

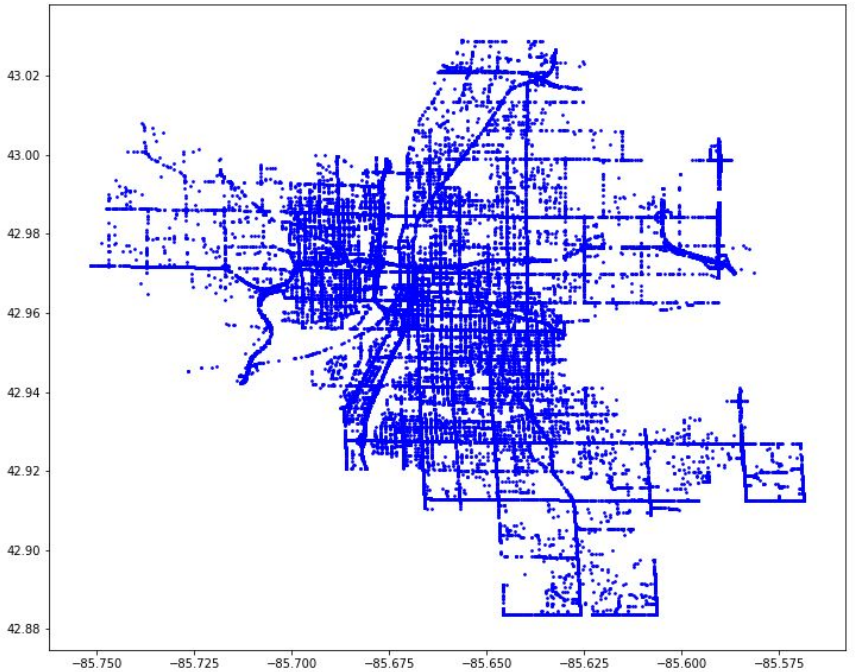
Explore the data:

	X	Y
0	-85.639647	42.927216
1	-85.639487	42.927213
2	-85.639387	42.927212
3	-85.639288	42.927210
4	-85.639288	42.927210
5	-85.639188	42.927208
6	-85.639168	42.927208
7	-85.639108	42.927207

Building an Original Model

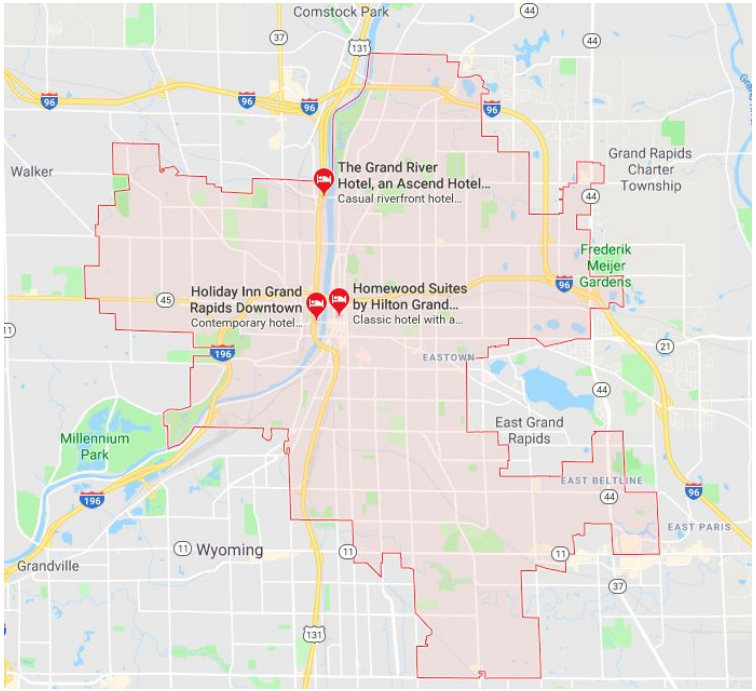
Explore the data:

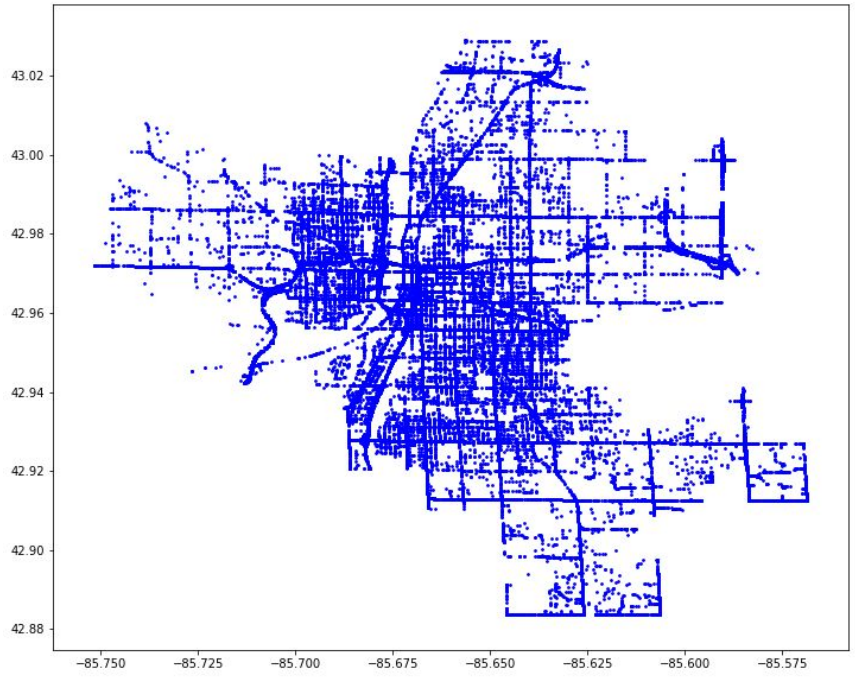
	X	Y
0	-85.639647	42.927216
1	-85.639487	42.927213
2	-85.639387	42.927212
3	-85.639288	42.927210
4	-85.639288	42.927210
5	-85.639188	42.927208
6	-85.639168	42.927208
7	-85.639108	42.927207



Building an Original Model

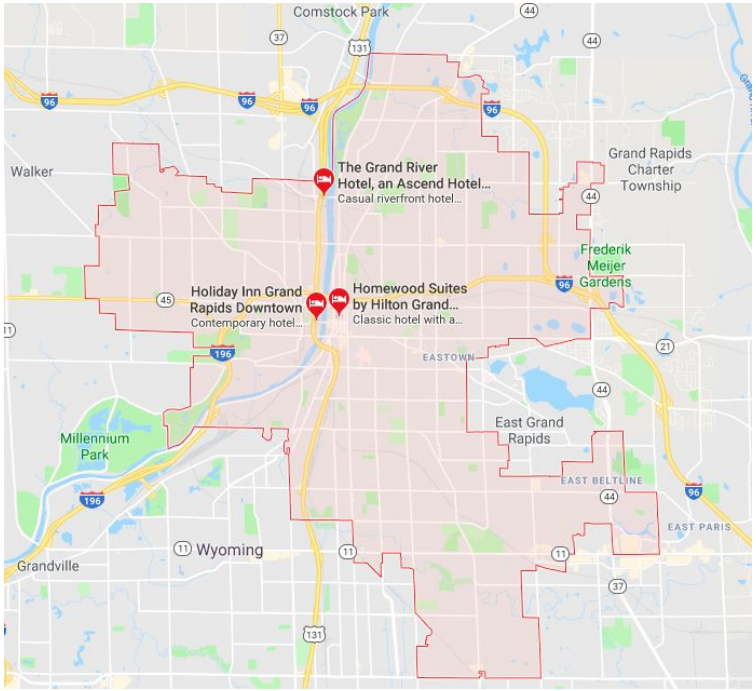
Explore the data:

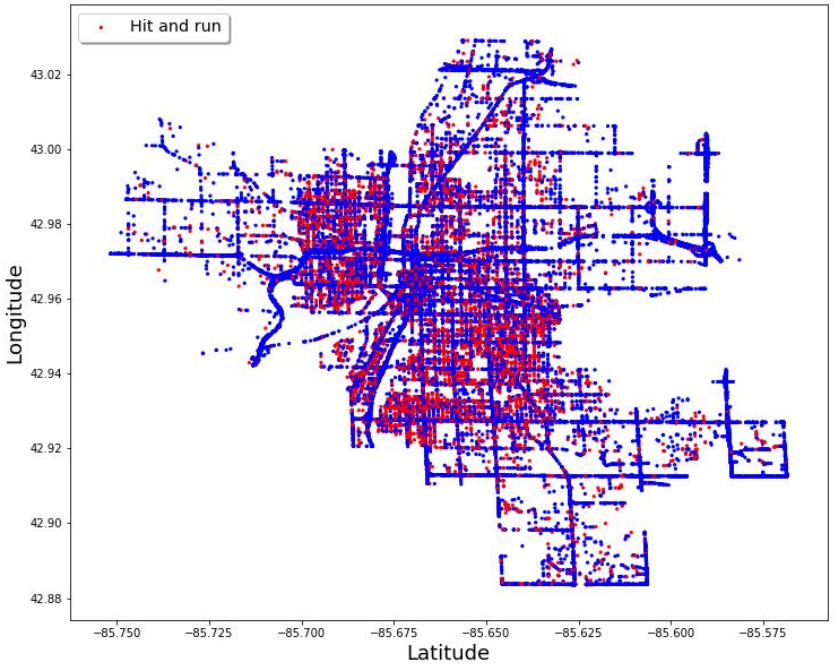




Building an Original Model

Explore the data:





Building an Original Model

Clean/transform the data:

```
In [91]: crash[['X', 'Y', 'DRIVER1AGE', 'NUMOFINJ']].head()
```

Out[91]:

	X	Y	DRIVER1AGE	NUMOFINJ
0	-85.639647	42.927216	62.0	0
1	-85.639487	42.927213	31.0	0
2	-85.639387	42.927212	22.0	0
3	-85.639288	42.927210	30.0	0
4	-85.639288	42.927210	44.0	0

Building an Original Model

Clean/transform the data:

In [34]:
```
sc_X = StandardScaler()
X_scaled = sc_X.fit_transform(crash[['X', 'Y', 'DRIVER1AGE', 'NUMOFINJ']])
X_scaled_df = pd.DataFrame(X_scaled, columns = ['X', 'Y', 'DRIVER1AGE', 'NUMOFINJ'])
crash_ = crash.drop(['X', 'Y', 'DRIVER1AGE', 'NUMOFINJ'], axis=1)
crash = pd.concat([X_scaled_df, crash_], axis=1)
crash.iloc[:, :4].head()
```

Out[34]:

	X	Y	DRIVER1AGE	NUMOFINJ
0	0.406318	-0.996140	1.685157	-0.416816
1	0.411006	-0.996237	-0.269018	-0.416816
2	0.413936	-0.996298	-0.836360	-0.416816
3	0.416866	-0.996358	-0.332056	-0.416816
4	0.416866	-0.996358	0.550474	-0.416816

Building an Original Model

Clean/transform the data:

In [111]:
```
crash[['CRASHSEVER', 'DRIVER1SEX','EMRGVEH', 'HITANDRUN',
       'MOTORCYCLE', 'D1COND','D1DRINKIN']].head()
```

Out[111]:

	CRASHSEVER	DRIVER1SEX	EMRGVEH	HITANDRUN	MOTORCYCLE	D1COND	D1DRINKIN
0	Property Damage Only	F	No	Yes	No	Appeared Normal	No
1	Property Damage Only	M	No	Yes	No	Unknown	No
2	Property Damage Only	F	No	No	No	Appeared Normal	No
3	Property Damage Only	M	No	Yes	No	Appeared Normal	No
4	Property Damage Only	M	No	No	No	Appeared Normal	No

Building an Original Model

Clean/transform the data:

```
In [135]: dummies.head()
```

Out[135]:

	CRASHSEVER_Fatal	CRASHSEVER_Injury	CRASHSEVER_Property Damage Only	DRIVER1SEX_F	DRIVER1SEX_M	DRIVER1SEX_U
0	0	0	1	1	0	0
1	0	0	1	0	1	0
2	0	0	1	1	0	0
3	0	0	1	0	1	0
4	0	0	1	0	1	0

5 rows × 21 columns

Testimonial from Mike!

ADAC

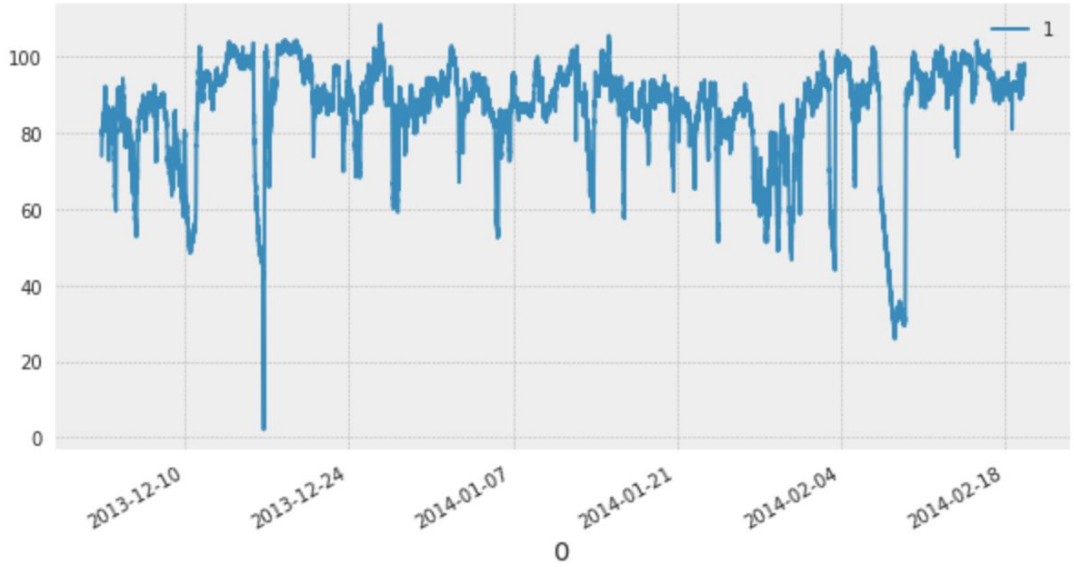
Building an Original Model

Tech stack:

Building an Original Model

AWS RCF demo:

2013-12-02 21:15:00	73.967322
2013-12-02 21:20:00	74.935882
2013-12-02 21:25:00	76.124162
2013-12-02 21:30:00	78.140707
2013-12-02 21:35:00	79.329836

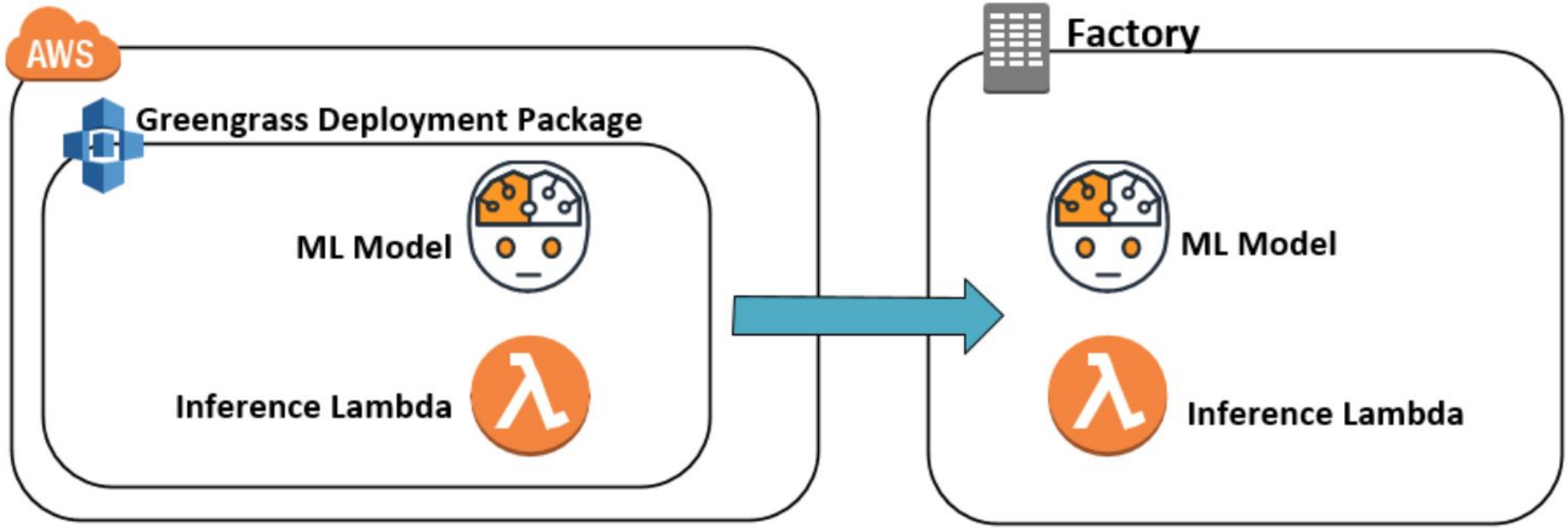


Deploying a Trained Model

All AWS in cloud:

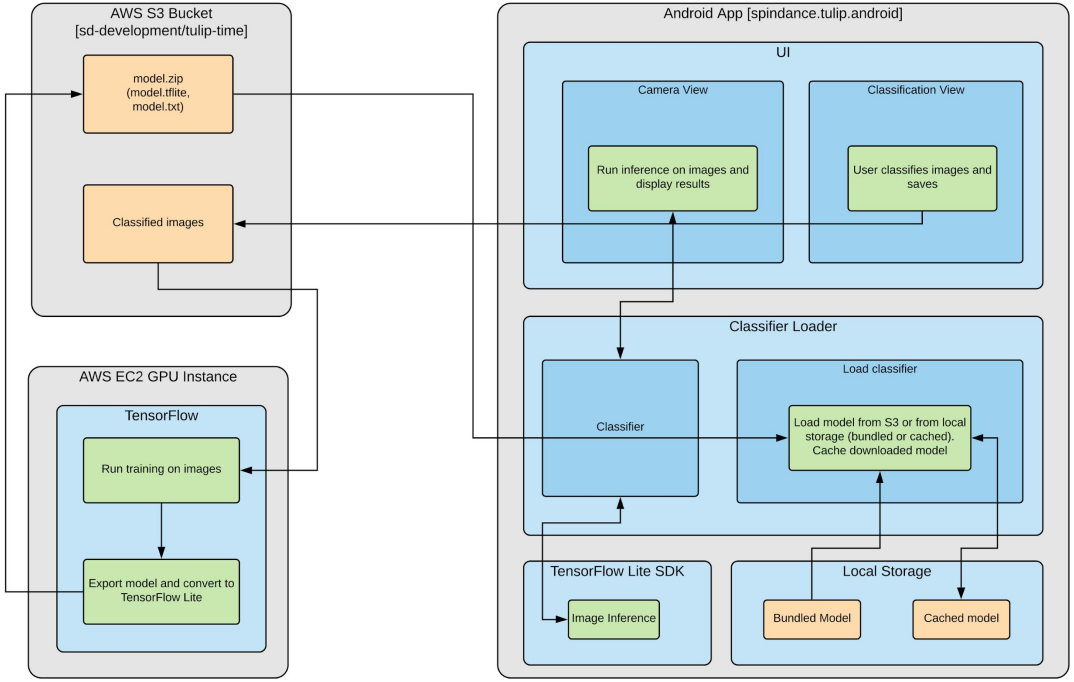
Deploying a Trained Model

All AWS on edge:



Deploying a Trained Model

Some AWS

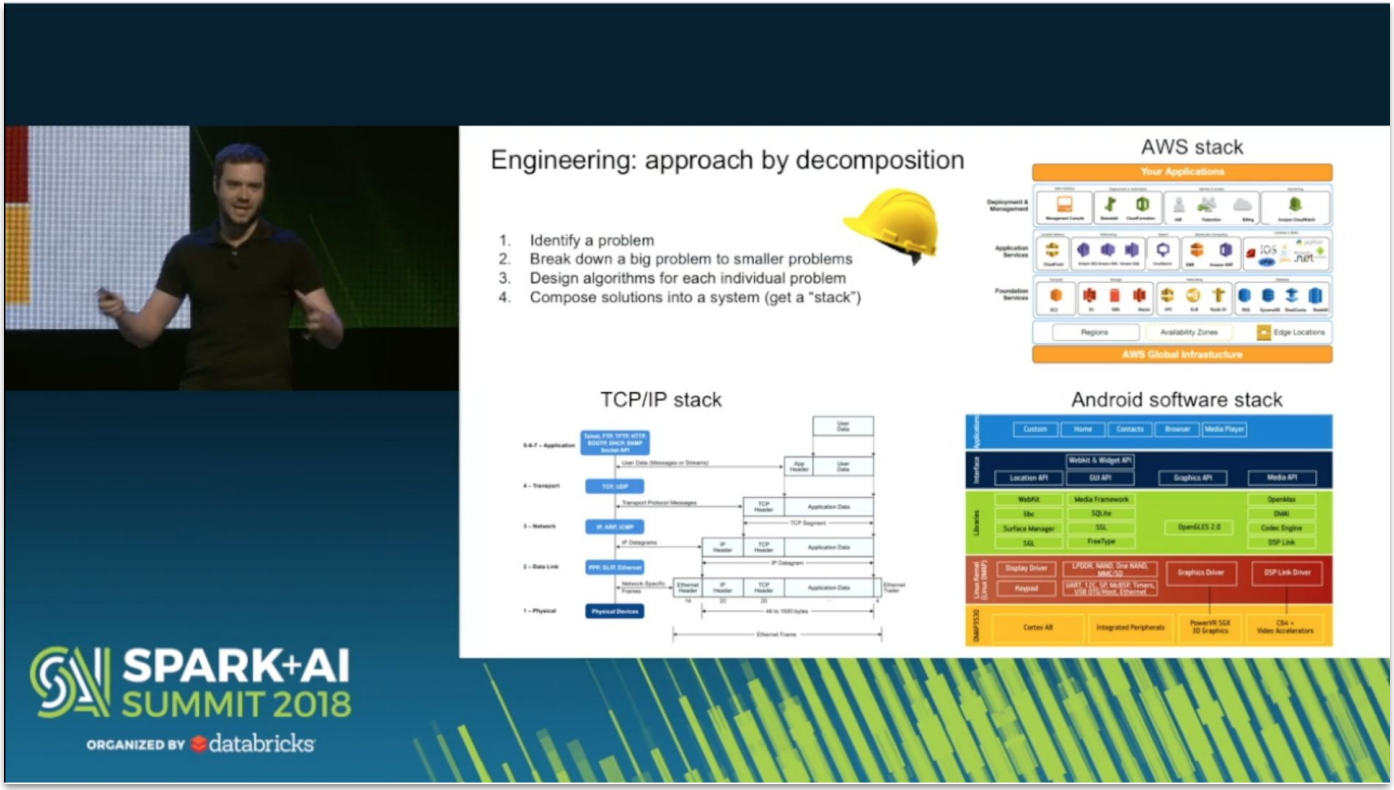


TensorFlow Lite

Deploying a Trained Model

No AWS

Software 2.0


Engineering: approach by decomposition
1. Identify a problem
2. Break down a big problem to smaller problems
3. Design algorithms for each individual problem
4. Compose solutions into a system (get a "stack")
AWS stack
Your Applications
Regions
Availability Zones
Edge Locations
AWS Global Infrastucture
TCP/IP stack
Android software stack
SPARK+AI SUMMIT 2018
ORGANIZED BY databricks