Global Big Data Conference

GLOBAL ARTIFICIAL INTELLIGENCE CONFERENCE

BOSTON

OCT 1st, 2nd, 3rd 2019

Boston Convention & Exhibition Center

www.globalbigdataconference.com

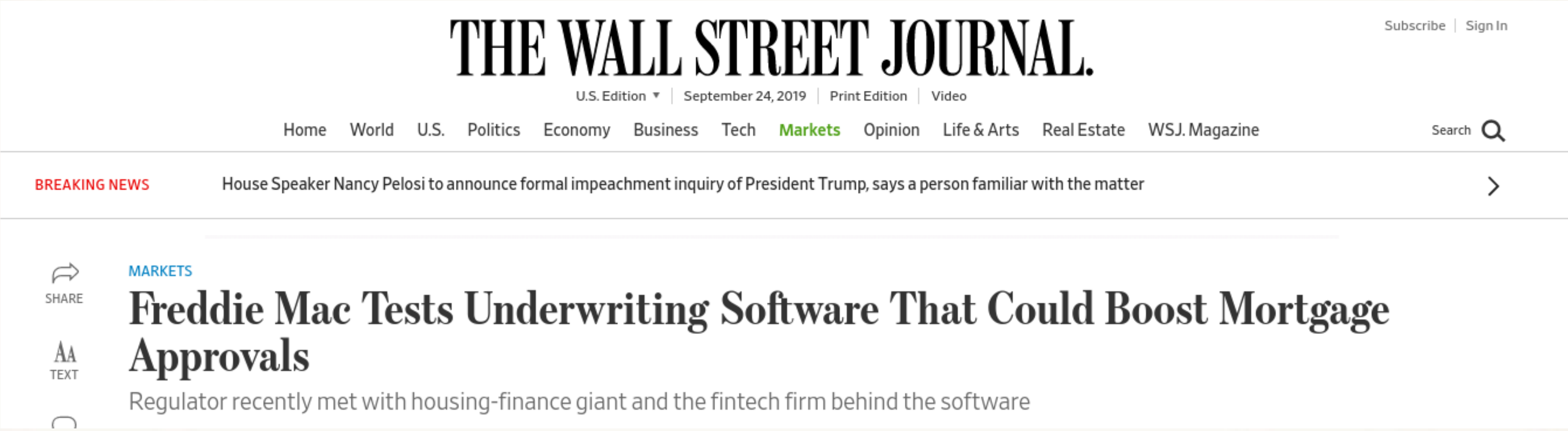
Twitter : @bigdataconf
#GAIC

Financial Loan Risk Analysis Using Keras Deep Learning

Marty Lurie, marty@cloudera.com

The business question: Can we use AI/Deep Learning to predict if a loan will be repaid? Specifically, can we predict if there will be foreclosure costs associated with a loan based on the features of the loan?

Breaking news September 24, 2019… (today is October 3, 2019)


Subscribe | Sign In
U.S. Edition ▼ | September 24, 2019 | Print Edition | Video
Home World U.S. Politics Economy Business Tech Markets Opinion Life & Arts Real Estate WSJ. Magazine
Search
SHARE
TEXT

THE WALL STREET JOURNAL.

BREAKING NEWS House Speaker Nancy Pelosi to announce formal impeachment inquiry of President Trump, says a person familiar with the matter

MARKETS

Freddie Mac Tests Underwriting Software That Could Boost Mortgage Approvals

Regulator recently met with housing-finance giant and the fintech firm behind the software

Agenda

- Statistics, Machine Learning, and AI, Oh My!
- Why Fannie Mae Loan Analysis?
- Architecture for ML/Deep Learning
- Data Engineering
- Logistic Regression
- Keras Neural Network
- Results and what’s next
- Here are the whole works: https://github.com/git4impatient/fanniemae

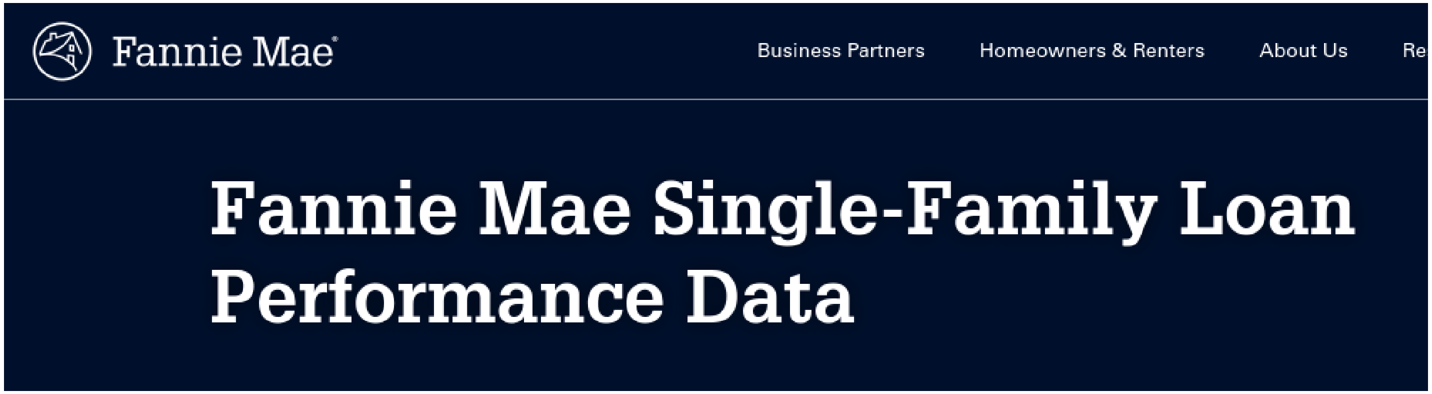
Statistics, Machine Learning, and AI, Oh My!

- ***Statistics*** is the discipline that concerns the collection, organization, displaying, analysis, interpretation and presentation of data.[1][2][3]
- ***Machine learning (ML)*** is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.
- In computer science, ***artificial intelligence (AI)***, sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans.
- All definitions from https://en.wikipedia.org

Anyone ever try to change a definition in wikipedia?

Why Fannie Mae Loan Analysis?

- The Federal National Mortgage Association (FNMA), commonly known as Fannie Mae, is a United States government-sponsored enterprise (GSE) and, since 1968, a publicly traded company. Founded in 1938 during the Great Depression as part of the New Deal,[2] the corporation's purpose is to expand the secondary mortgage market by securitizing mortgage loans in the form of mortgage-backed securities (MBS),[3] *Source: wikipedia.org*
- This talk is based on data from: The Home Affordable Refinance Program (HARP) is a federal refinance program targeting underwater homeowners. *https://www.hsh.com/finance/refinance/what-is-harp-do-i-qualify-for-a-harp-loan.html*



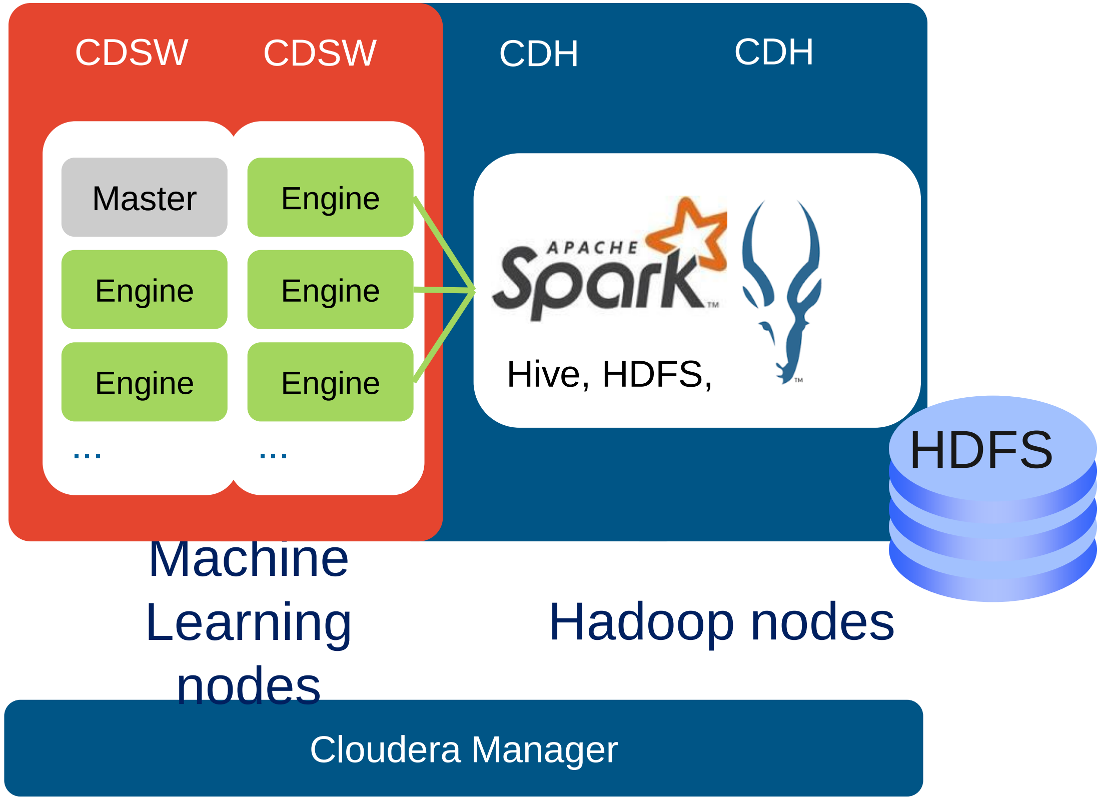
Fannie Mae Single-Family Loan Performance Data

https://www.fanniemae.com/portal/funding-the-market/data/loan-performance-data.html

Architecture for ML/Deep Learning

Projects/Software Used:

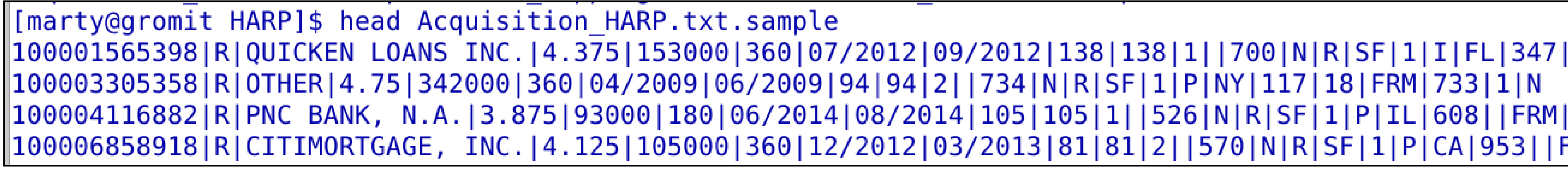
- Apache Spark
- Apache Impala
- HUE
- Apache Hadoop
- Tensorflow
- Keras
- Pandas
- Numpy
- Seaborn
- Matplotlib
- CDSW



© Cloudera, Inc. All rights reserved.

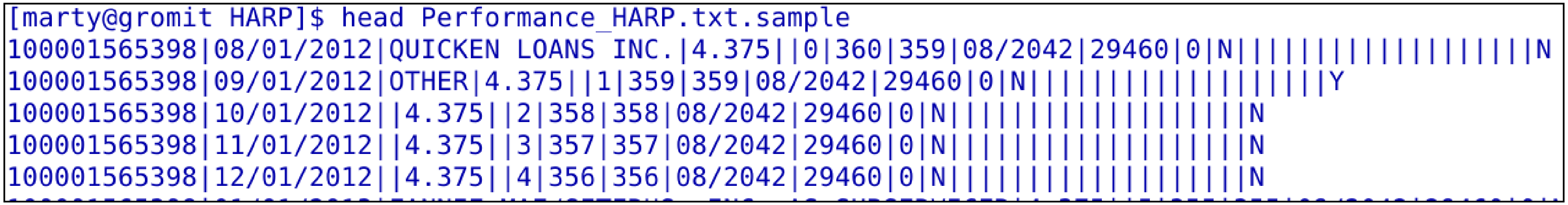
Data Engineering

- “80% of analysts’ time is spent discovering and preparing data” *Harvard Business Review*
- Source data: Fannie Mae HARP
  - Acquisition file: loan origination details



```
[marty@gromit HARP]$ head Acquisition_HARP.txt.sample
100001565398|R|QUICKEN LOANS INC.|4.375|153000|360|07/2012|09/2012|138|138|1||700|N|R|SF|1|I|FL|347|
100003305358|R|OTHER|4.75|342000|360|04/2009|06/2009|94|94|2||734|N|R|SF|1|P|NY|117|18|FRM|733|1|N
100004116882|R|PNC BANK, N.A.|3.875|93000|180|06/2014|08/2014|105|105|1||526|N|R|SF|1|P|IL|608||FRM|
100006858918|R|CITIMORTGAGE, INC.|4.125|105000|360|12/2012|03/2013|81|81|2||570|N|R|SF|1|P|CA|953||F
```

  - Performance file: payment status, repeating rows



```
[marty@gromit HARP]$ head Performance_HARP.txt.sample
100001565398|08/01/2012|QUICKEN LOANS INC.|4.375||0|360|359|08/2042|29460|0|N|||||||||||||||||||N
100001565398|09/01/2012|OTHER|4.375||1|359|359|08/2042|29460|0|N|||||||||||||||||||Y
100001565398|10/01/2012||4.375||2|358|358|08/2042|29460|0|N|||||||||||||||||||N
100001565398|11/01/2012||4.375||3|357|357|08/2042|29460|0|N|||||||||||||||||||N
100001565398|12/01/2012||4.375||4|356|356|08/2042|29460|0|N|||||||||||||||||||N
```

Data Engineering – Create Tables in Impala

```
use fanniemae;
drop table if exists loan_perf;
create external table loan_perf(
loan_identifier string,
monthly_reporting_period string,
servicer_name string,
current_interest_rate decimal ( 14,10 ) yada yada yada
… lines deleted …
foreclosure_principal_write_off_amount decimal ( 11,2 ),
servicing_activity_indicator string
)
row format delimited fields terminated by '|'
stored as textfile location '/user/marty/fanniemae/perf'
;

select count(*) from loan_perf;
select * from loan_perf limit 1;
```

Data Engineering – SQL w/Impala Example

- SparkML, Tensorflow require numeric variables, best if scaled to values 0 to 1

```
drop table if exists loanacq_sqlnormed_p ;
create table loanacq_sqlnormed_p stored as parquet as
select loan_identifier,
case origination_channel
when 'R' then .1
when 'C' then .2
when 'B' then .3
end channel,
seller_name,
original_interest_rate/7.75 intrate,
original_upb/1402000.0 loanamt,
original_loan_to_value/ 97 loan2val,
number_of_borrowers/6 numborrowers,
--original_debt_to_income_ratio/64 debt2income,
borrower_credit_score_at_origination/842 creditscore,
property_state,
origination_date
from loan_acquisition;
```

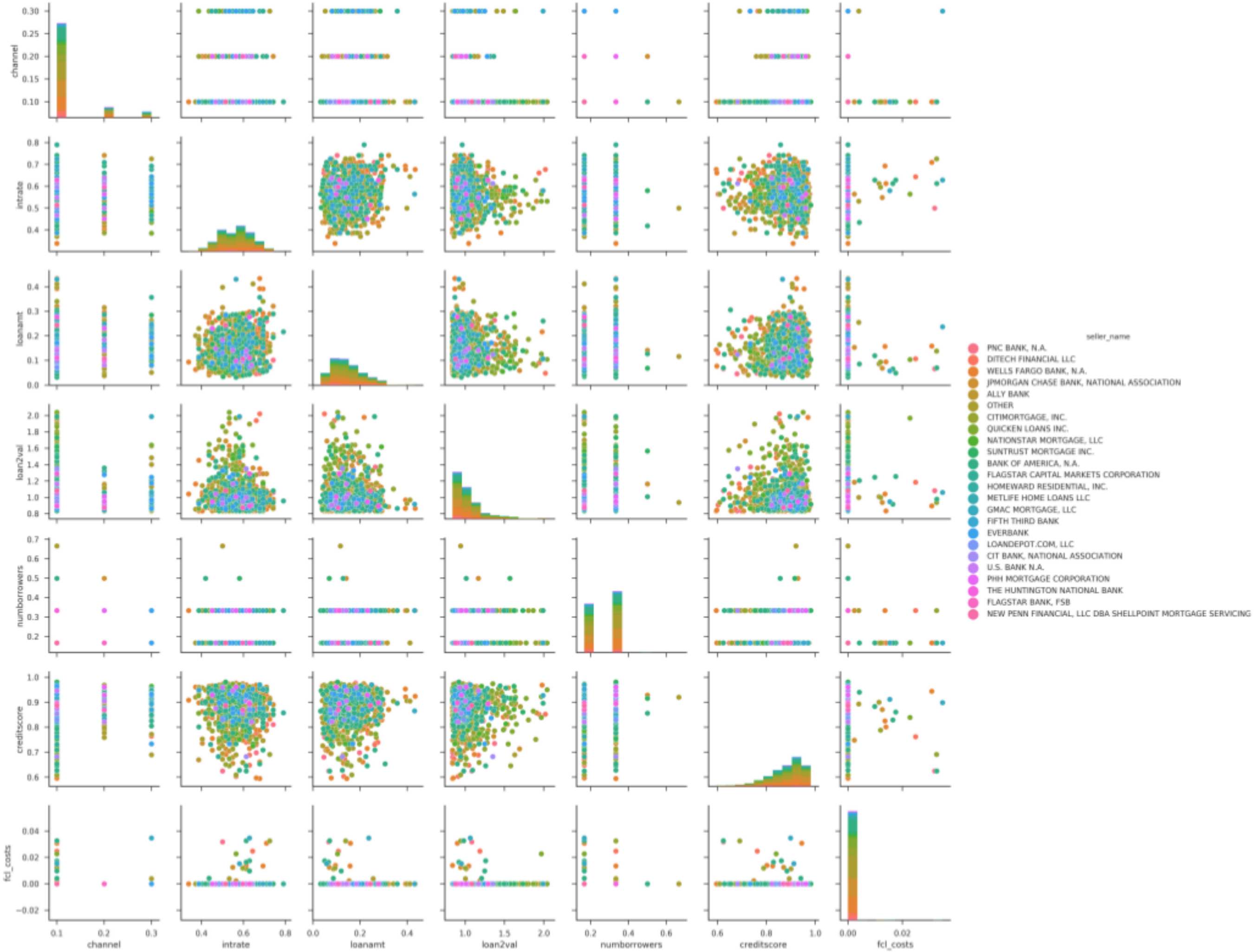
Data Engineering – pySpark Example

- SparkML, Tensorflow require numeric variables, best if scaled to values 0 to 1

```
# need to convert from text field to numeric
# common requirement when using sparkML
from pyspark.ml.feature import StringIndexer
# this will convert each unique
# string into a numeric
indexer = \
StringIndexer(inputCol="property_state", \
outputCol="loc_state")
indexed = indexer.fit(lndf).transform(lndf)
indexed.show(5)
```

Data Engineering – Know Your Data
seaborn pairplot

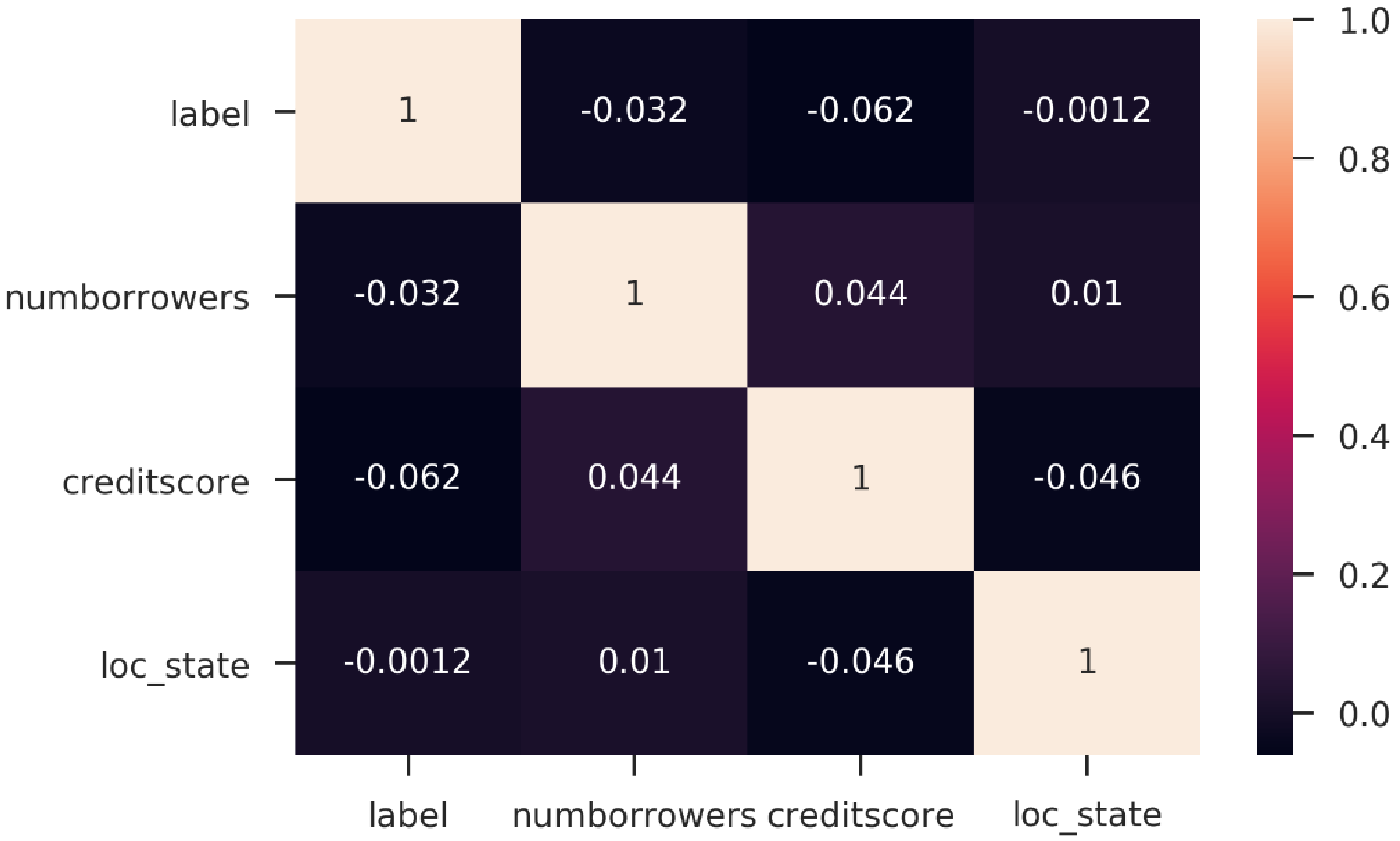
```
g = sns.pairplot(pdsdf,  hue="seller_name" )
```



Data Engineering – Know Your Data seaborn heatmap

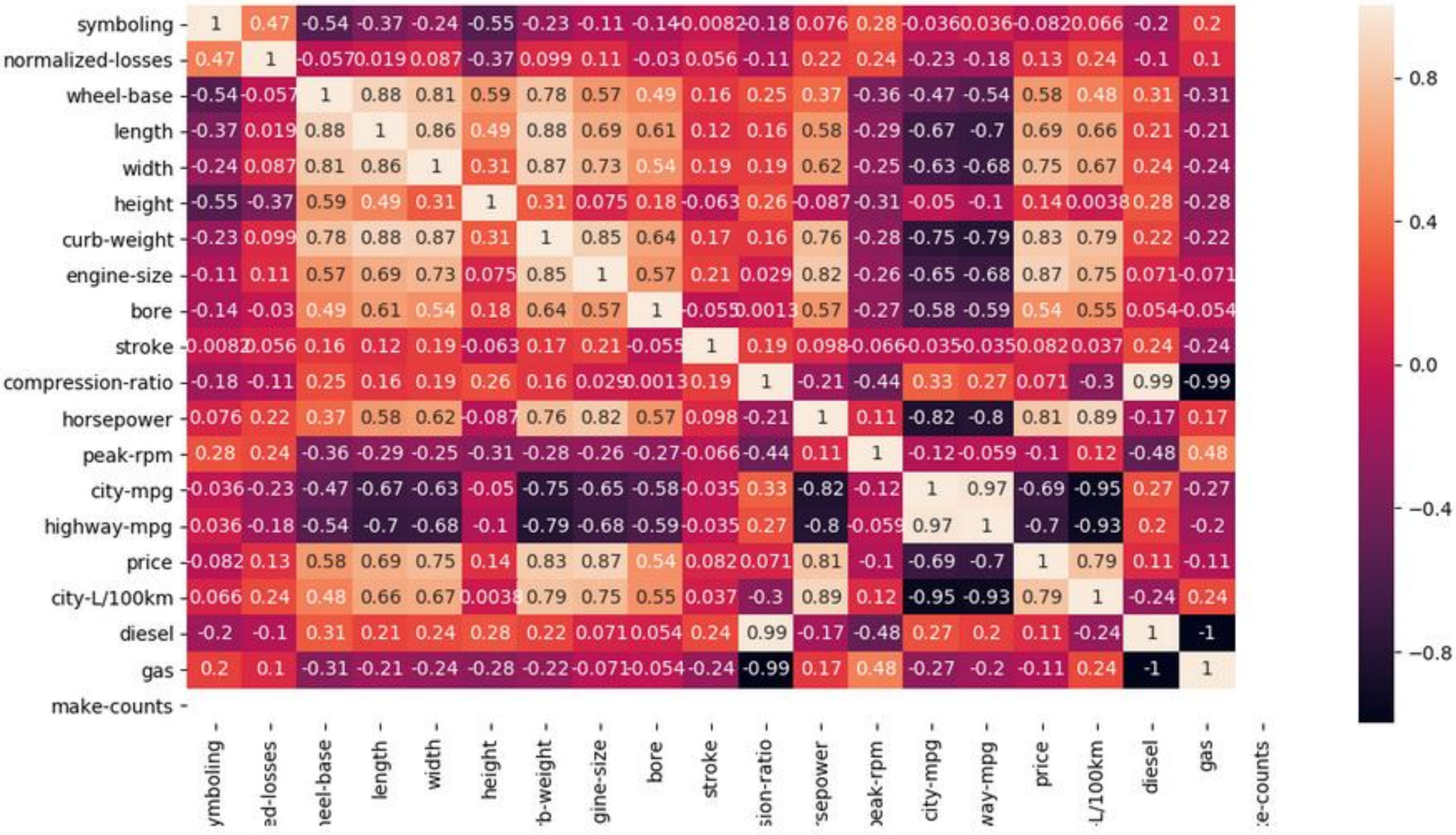
```
sns.heatmap(kerasinputpsdf.corr(), annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6622cda310>
```



Data Engineering – Know Your Data Example with stronger correlations

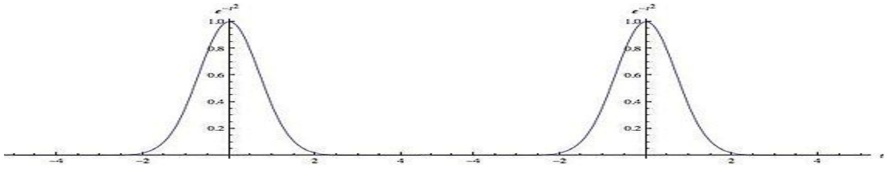
http://www.codeheroku.com/post.html?name=Introduction%20to%20Exploratory%20Data%20Analysis%20(EDA)



	ymboling	ed-losses	neel-base	length	width	height	b-weight	gine-size	bore	stroke	sion-ratio	sepower	eak-rpm	city-mpg	way-mpg	price	L/100km	diesel	gas	e-counts
symboling	1	0.47	-0.54	-0.37	-0.24	-0.55	-0.23	-0.11	-0.14	-0.0082	-0.18	0.076	0.28	-0.036	0.036	-0.082	0.066	-0.2	0.2	
normalized-losses	0.47	1	-0.057	0.019	0.087	-0.37	0.099	0.11	-0.03	0.056	-0.11	0.22	0.24	-0.23	-0.18	0.13	0.24	-0.1	0.1	
wheel-base	-0.54	-0.057	1	0.88	0.81	0.59	0.78	0.57	0.49	0.16	0.25	0.37	-0.36	-0.47	-0.54	0.58	0.48	0.31	-0.31	
length	-0.37	0.019	0.88	1	0.86	0.49	0.88	0.69	0.61	0.12	0.16	0.58	-0.29	-0.67	-0.7	0.69	0.66	0.21	-0.21	
width	-0.24	0.087	0.81	0.86	1	0.31	0.87	0.73	0.54	0.19	0.19	0.62	-0.25	-0.63	-0.68	0.75	0.67	0.24	-0.24	
height	-0.55	-0.37	0.59	0.49	0.31	1	0.31	0.075	0.18	-0.063	0.26	-0.087	-0.31	-0.05	-0.1	0.14	0.0038	0.28	-0.28	
curb-weight	-0.23	0.099	0.78	0.88	0.87	0.31	1	0.85	0.64	0.17	0.16	0.76	-0.28	-0.75	-0.79	0.83	0.79	0.22	-0.22	
engine-size	-0.11	0.11	0.57	0.69	0.73	0.075	0.85	1	0.57	0.21	0.029	0.82	-0.26	-0.65	-0.68	0.87	0.75	0.071	-0.071	
bore	-0.14	-0.03	0.49	0.61	0.54	0.18	0.64	0.57	1	-0.055	0.0013	0.57	-0.27	-0.58	-0.59	0.54	0.55	0.054	-0.054	
stroke	-0.0082	0.056	0.16	0.12	0.19	-0.063	0.17	0.21	-0.055	1	0.19	0.098	-0.066	-0.035	-0.035	0.082	0.037	0.24	-0.24	
compression-ratio	-0.18	-0.11	0.25	0.16	0.19	0.26	0.16	0.029	0.0013	0.19	1	-0.21	-0.44	0.33	0.27	0.071	-0.3	0.99	-0.99	
horsepower	0.076	0.22	0.37	0.58	0.62	-0.087	0.76	0.82	0.57	0.098	-0.21	1	0.11	-0.82	-0.8	0.81	0.89	-0.17	0.17	
peak-rpm	0.28	0.24	-0.36	-0.29	-0.25	-0.31	-0.28	-0.26	-0.27	-0.066	-0.44	0.11	1	-0.12	-0.059	-0.1	0.12	-0.48	0.48	
city-mpg	-0.036	-0.23	-0.47	-0.67	-0.63	-0.05	-0.75	-0.65	-0.58	-0.035	0.33	-0.82	-0.12	1	0.97	-0.69	-0.95	0.27	-0.27	
highway-mpg	-0.036	-0.18	-0.54	-0.7	-0.68	-0.1	-0.79	-0.68	-0.59	-0.035	0.27	-0.8	-0.059	0.97	1	-0.7	-0.93	0.2	-0.2	
price	-0.082	0.13	0.58	0.69	0.75	0.14	0.83	0.87	0.54	0.082	0.071	0.81	-0.1	-0.69	-0.7	1	0.79	0.11	-0.11	
city-L/100km	0.066	0.24	0.48	0.66	0.67	0.0038	0.79	0.75	0.55	0.037	-0.3	0.89	0.12	-0.95	-0.93	0.79	1	-0.24	0.24	
diesel	-0.2	-0.1	0.31	0.21	0.24	0.28	0.22	0.071	0.054	0.24	0.99	-0.17	-0.48	0.27	0.2	0.11	-0.24	1	-1	
gas	0.2	0.1	-0.31	-0.21	-0.24	-0.28	-0.22	-0.071	-0.054	-0.24	-0.99	0.17	0.48	-0.27	-0.2	-0.11	0.24	-1	1	
make-counts																				

Logistic Regression (Classification)

- “*Logistic regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary).*” https://www.statisticssolutions.com/what-is-logistic-regression/
- Two different Fannie Mae loan distributions, one with Foreclosure Costs one without Foreclosure costs:



- Does this help?

$$\text{logit}(p) = \log\left(\frac{p(y=1)}{1-(p=1)}\right) = \beta_0 + \beta_1 \cdot x_{i2} + \beta_2 \cdot x_{i2} + \_ + \beta_p \cdot x_{in}$$

for i = 1...n .

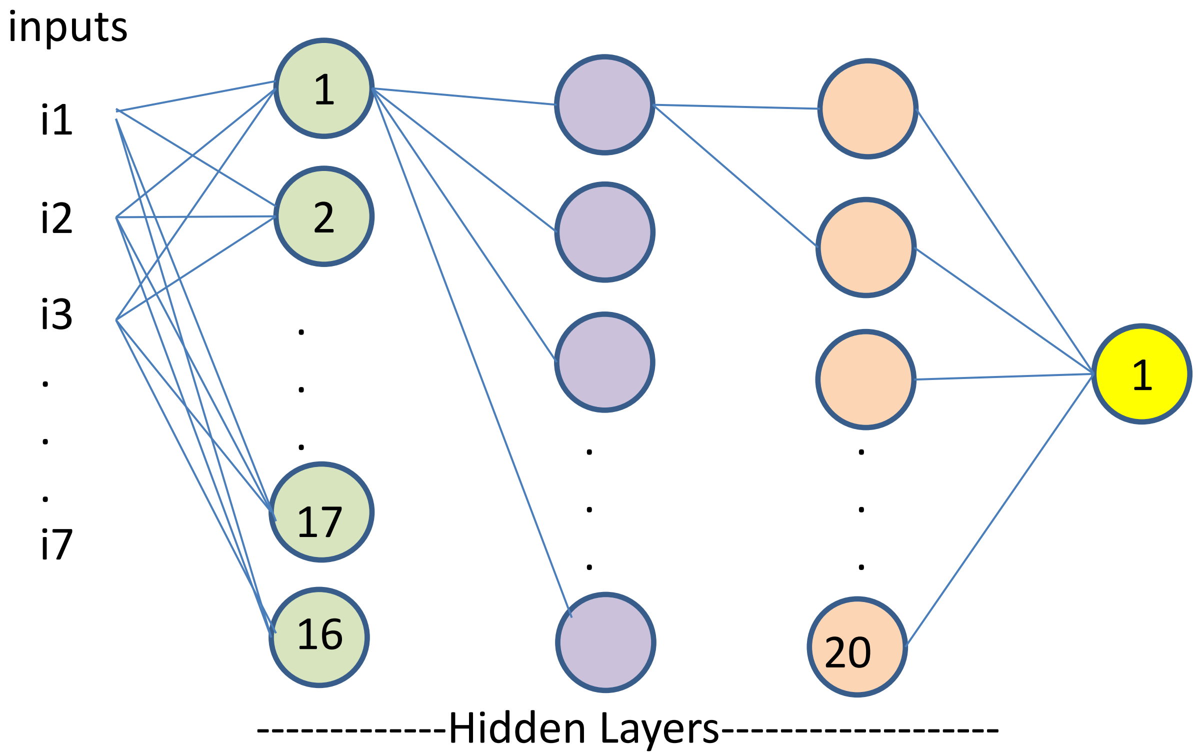
Logistic Regression

- pySpark Logistic Regression Inputs:
  - spark dataframe with: label and features

LABEL	FEATURES
0 OR 1 for foreclosure costs	[income, loan value, interest rate, etc]

```
# Create a LogisticRegression instance
lr = LogisticRegression(maxIter=10, regParam=0.01)
# Print out the parameters, documentation, and any default values.
print("LogisticRegression parameters:\n" + lr.explainParams() + "\n")
# Learn a LogisticRegression model. This uses the parameters stored in lr.
#  “output” is a lousy name for the input-dataframe, sorry ☹
model1 = lr.fit(output)
```

Keras Neural Network



Single output neuron

Not all connections show

Keras Neural Network

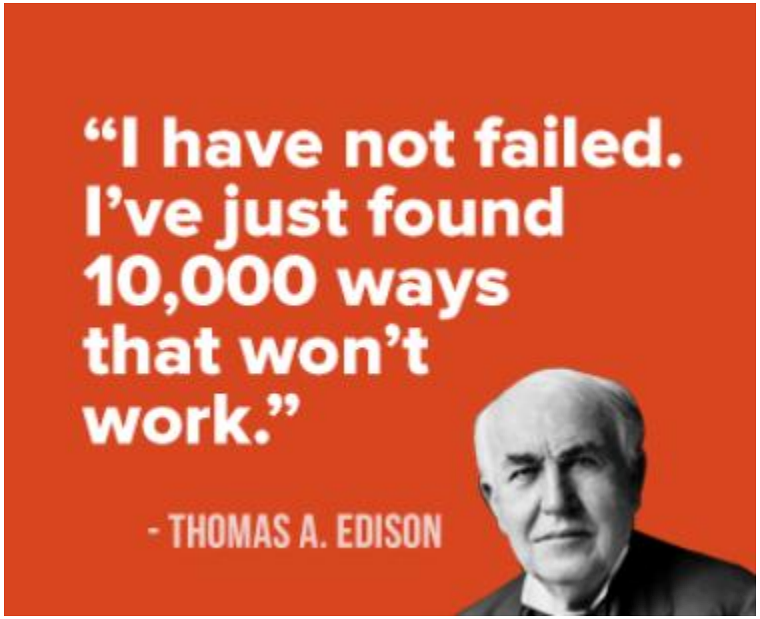
```
classifier = Sequential()
#First Hidden Layer
classifier.add(Dense(16, activation='relu', kernel_initializer='random_normal', input_dim=7))
#Second  Hidden Layer
classifier.add(Dense(20, activation='relu', kernel_initializer='random_normal'))
#Output Layer
classifier.add(Dense(1, activation='sigmoid', kernel_initializer='random_normal'))
#Compiling the neural network
classifier.compile(optimizer ='adam',loss='binary_crossentropy', metrics =['accuracy'])
classifier.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 16)                128

_________________________________________________________________
dense_2 (Dense)              (None, 20)                340

_________________________________________________________________
dense_3 (Dense)              (None, 1)                 21
=================================================================
Total params: 489
Trainable params: 489
Non-trainable params: 0
```

Results



“I have not failed. I’ve just found 10,000 ways that won’t work.”

- THOMAS A. EDISON

Results – Confusion Matrix

- Baseline Probability, incidence of foreclosure costs: 0.0213
- Logistic Regression, areaUnderROC: 0.738:

All Data (easy case)	Predict 0	Predict 1
Expect 0	502924	0
Expect 1	10942	0

Non-Diagonal is error

Diagonal is good

- Keras Deep Learni Prob of osts 2183/100571=0.0217

Training Data	Predi	Predict 1
Expect 0	402353	0
Expect 1	69	8671

Testing Data	Predic	Predict 1
Expect 0	1005	
Expect 1	19	3

Looked really good BUT the value of foreclosure costs was provided in the input matrix!

Results – Confusion Matrix

- Baseline Probability, incidence of foreclosure costs: 0.0213
- Logistic Regression, areaUnderROC: 0.738:

All Data (easy case)	Predict 0	Predict 1
Expect 0	502924	0
Expect 1	10942	0

Non-Diagonal is error

Diagonal is good

- Keras Deep Learning

Testing Data	Predict 0	Predict 1
Expect 0	97586	2985
Expect 1	1917	285

Wow we have a model!

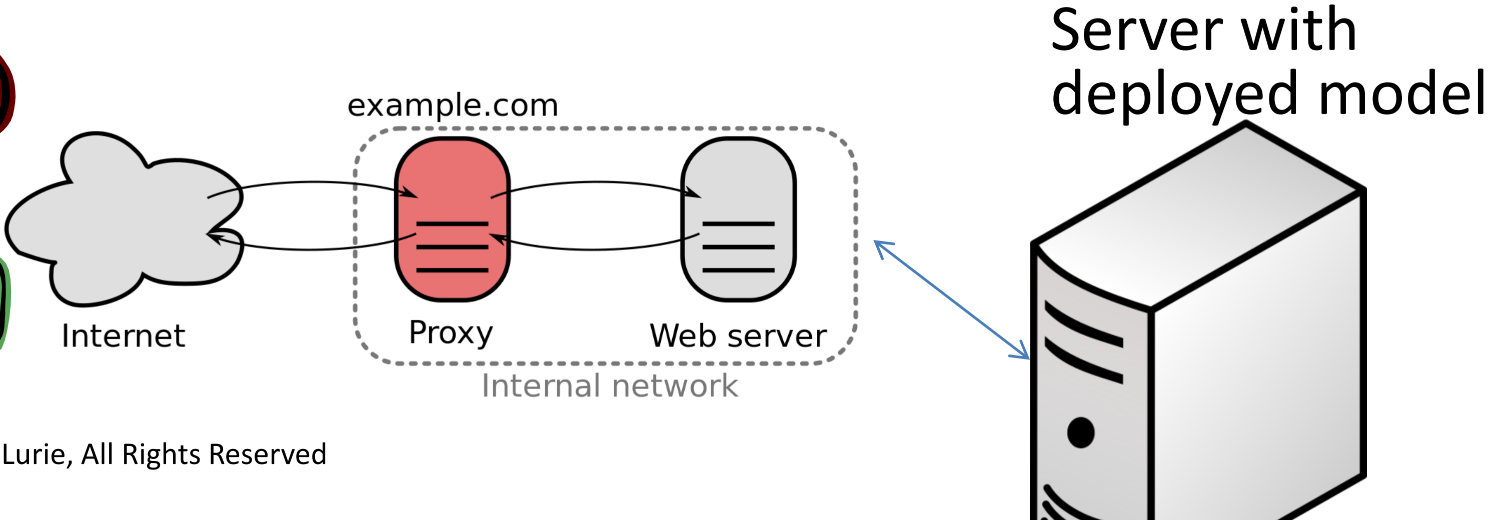
Are we done?

Not Yet…

What’s Next?

- We have a model, how do we use it? RestAPI

```
[marty@gromit fanniemae]$ curl -H "Content-Type: application/json" -X POST http://cdsw2.lurie.biz/api/altus-ds-1/models/call-model -d '{"accessKey":"mz7j447kq5q22o1vf4877fpdm6gl4n70","request":{"channel":0.1,"intrate":0.4,"loanamt":0.18,"loan2val":0.9,"numborrowers":0.33,"creditscore":0.91}}'
{
 "success": true,
 "response": "False"
}
```



What’s Next?

- Add more features, enrich data with external datasets
- Experiment with Hyperparameters: layers, neurons etc
- TensorBoard for model validation
- Model deployment – restAPI
- Full FannieMae dataset
  - HARP is subset of data that fits on my laptop
  - Some columns ignored that may improve model

Thank You!
Questions?

marty@cloudera.com