(つ◕ᴗ◕)つ━☆ﾟ.*･｡ﾟ

Intro to Machine Learning

Gianluca Segato – founder @ Uniwhere

Social handles: @giansegato

uniwhere

Agenda

stuff we’re going to talk about

Agenda

- Context & Speaker
- What are we talking about? What is Machine Learning?
- How we got here? What is this hype all about?
- Let’s dive a little bit deeper:
  - Supervised
  - Unsupervised
  - Recommendations systems

Agenda / 2

- Typical pipeline
- Deep Learning
- A couple of take-aways:
  - The concept of generalization
  - In the end, it all comes down to data
- Let’s see some code!
- Conclusions & Q&A

Context & Speaker

ie. who’s this beardless guy?

Context & Speaker

- Economics & Finance
- Uniwhere: a career manager: Berlin <-> Padua
- ML & AI since a couple of years
- Clue Data Hackathon and other stuff

What is Machine Learning?

ie. a layman (kind of) definition

Machine Learning

Computer can learn without being explicitly programmed.
(and change behavior when exposed to new data)

The intuition

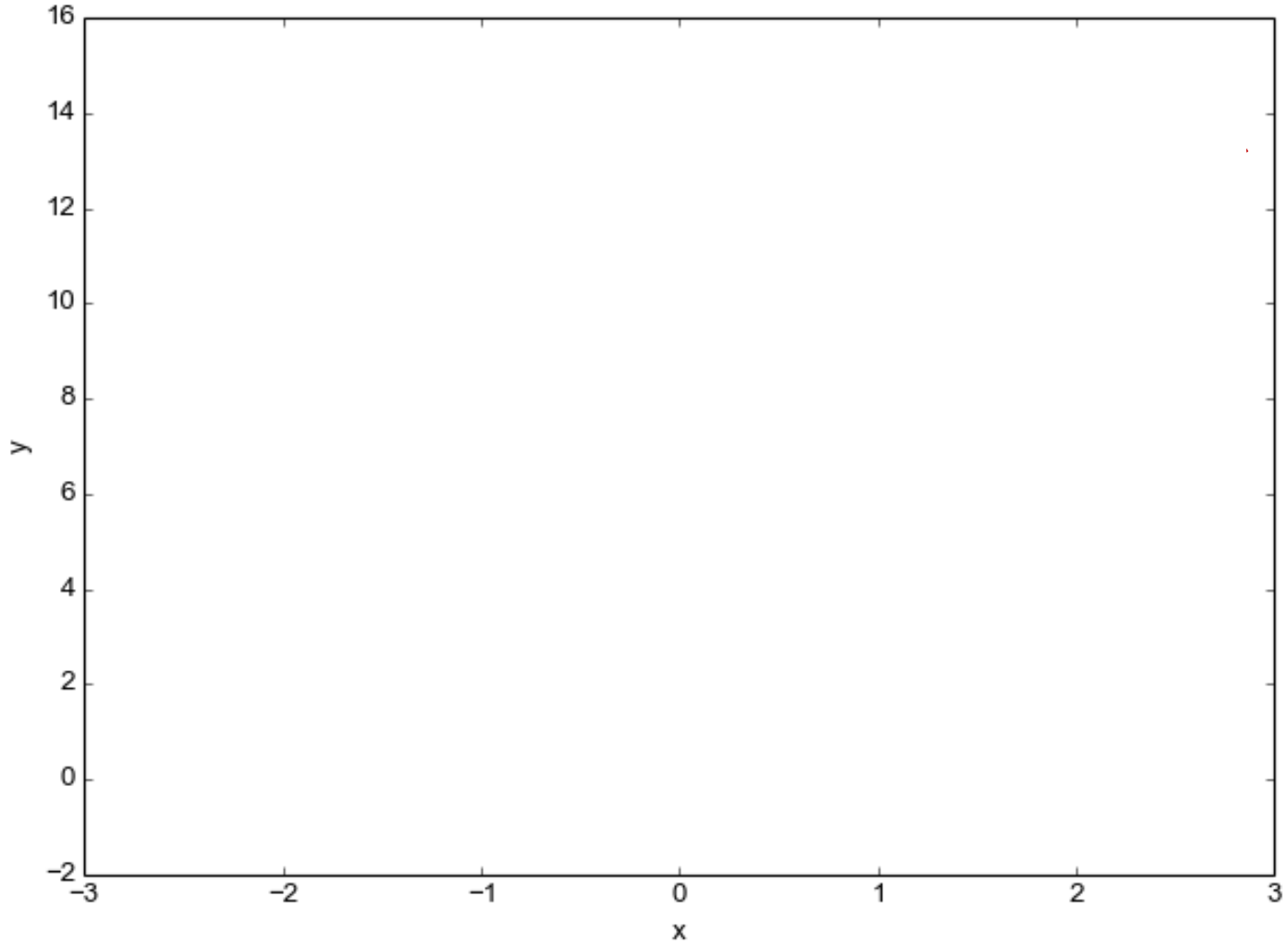
- Say, you want to develop an algorithm that’s able to predict how many people will attend the next GDG DevFest Veneto

- Programmatic approach:
  - Define the baseline: let’s assume 100 people (the room’s capacity)
  - If Friday: -20%
  - If Google is trending on Twitter: +15%
  - [+2π * (Alphabet stocks gain in the last week)]%
  - If sunny day: -10%
  - Return output

- It’s not gonna work :) let’s take a look at past attendance instead...

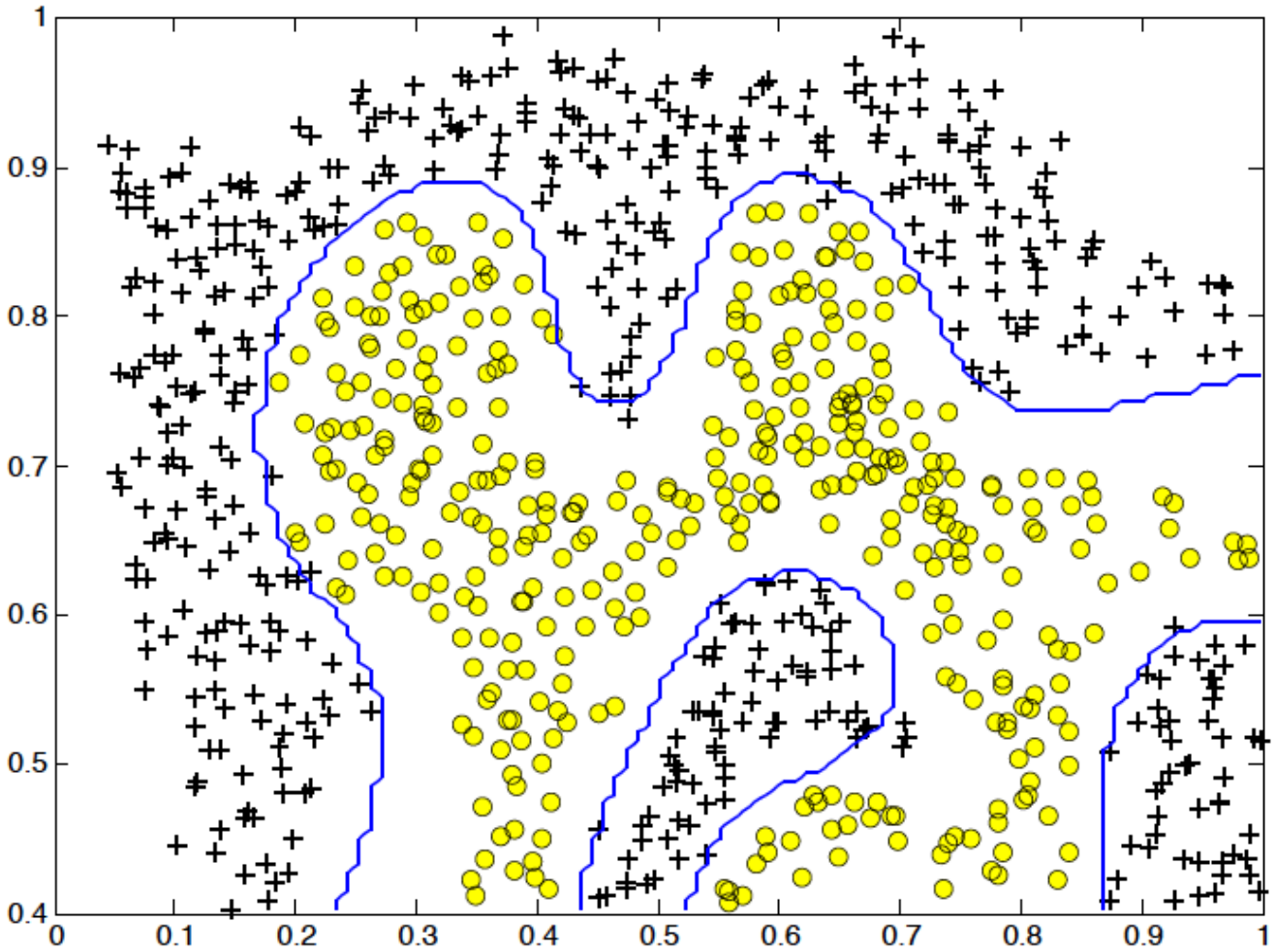
The Intuition

People attending (target variable, dependent variable, y)

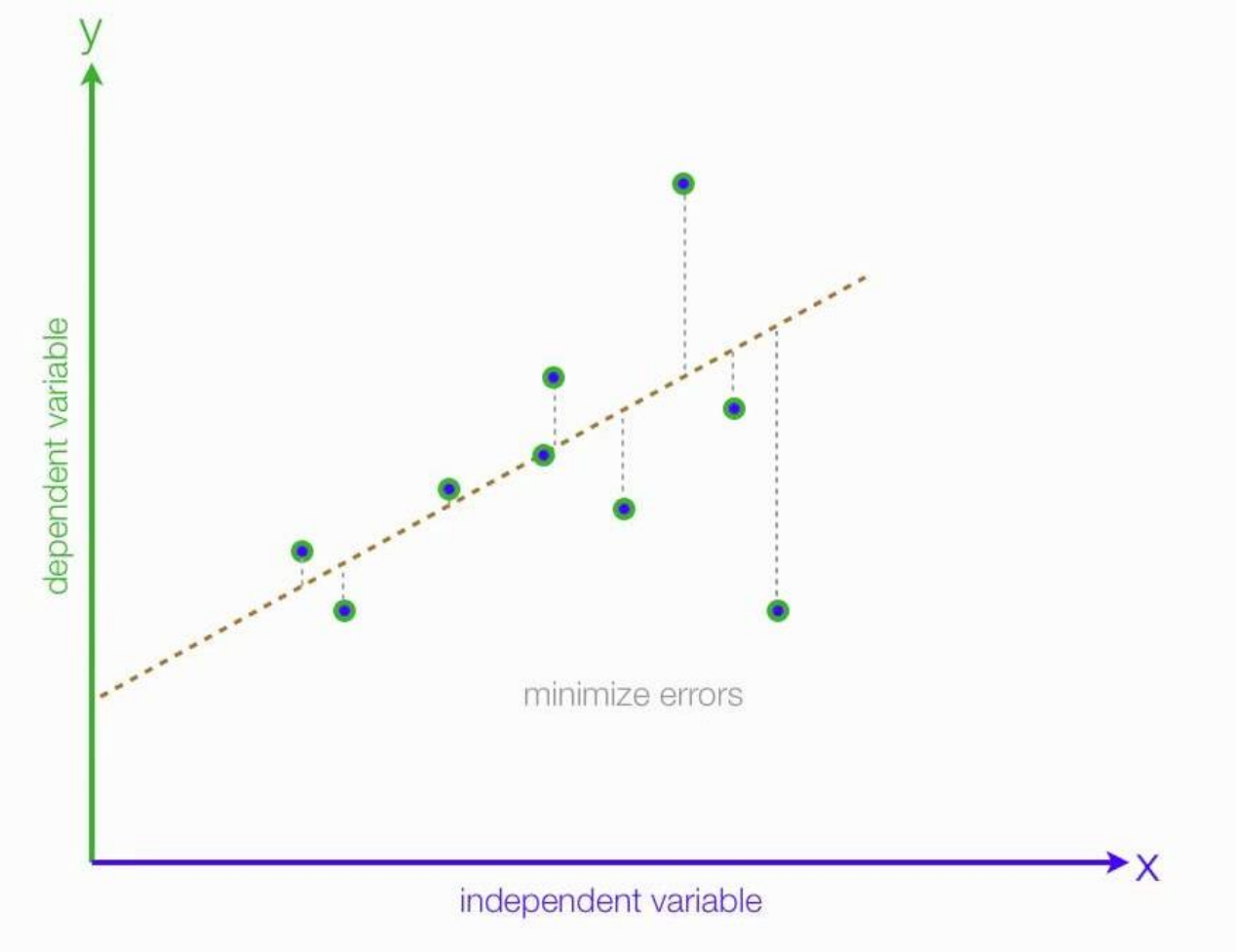


Bottles of beers present at the last event (features, independent variable, x, X)

The Intuition



The Intuition



$$f(x) = \hat{y} = \beta_0 + \beta_1 x$$

$$\text{Find } \min_{\beta_0, \beta_1} \mathrm{C}(\beta_0, \beta_1) \text{ for}$$

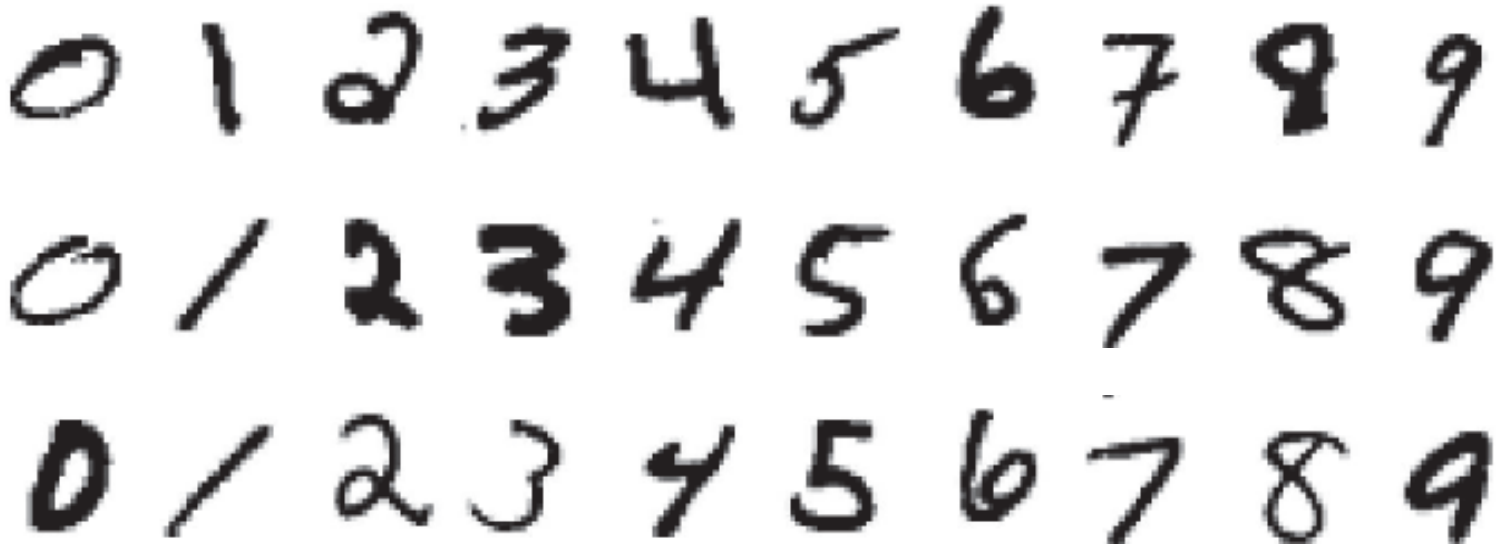
$$\mathrm{C}(\beta_0, \beta_1) = \sum (f(x_i) - y_i)^2$$

$$= \sum (y - \beta_0 - \beta_1 x_i)^2$$

Machine Learning

Computer can learn without being explicitly programmed.

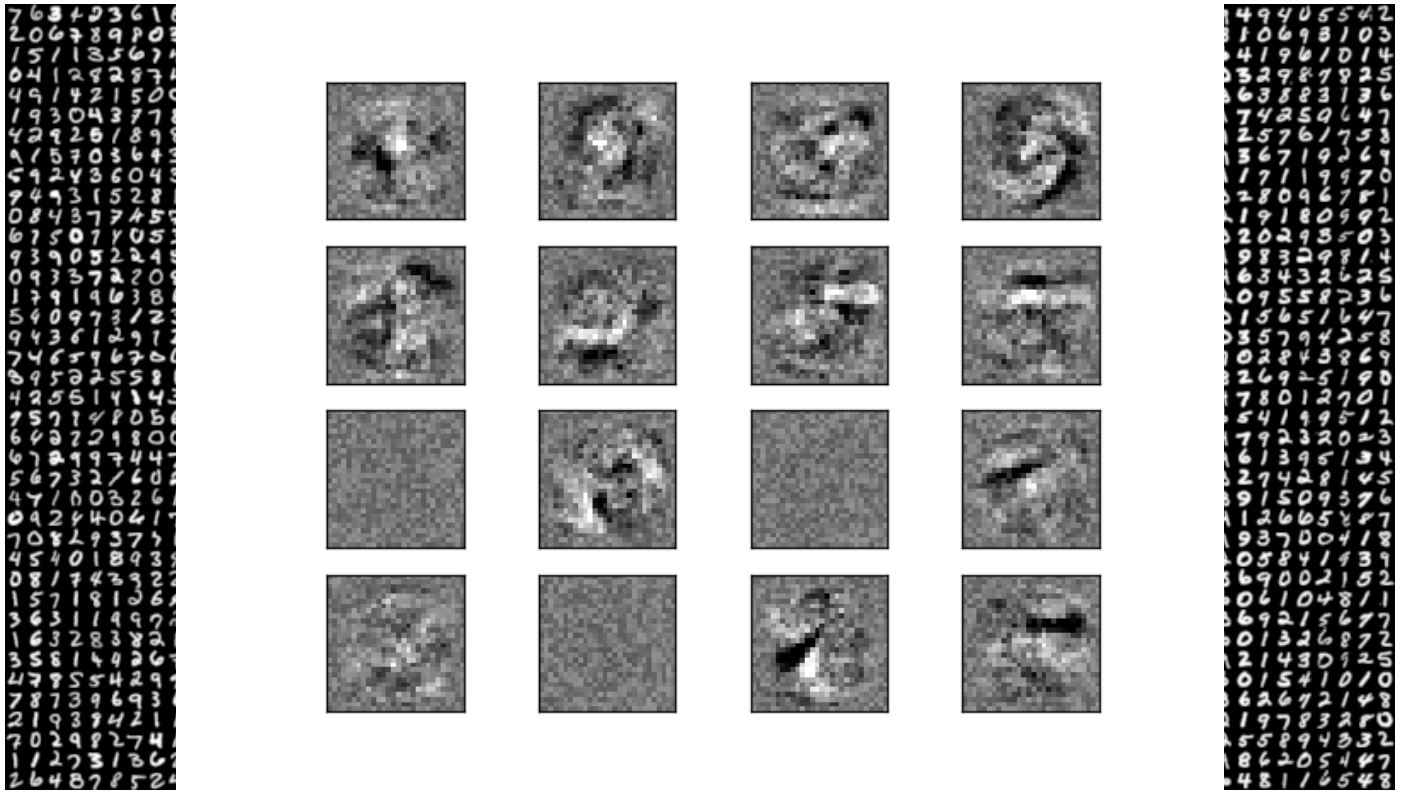
(and change behavior when exposed to new data)



Machine Learning

Computer can learn without being explicitly programmed.

(and change behavior when exposed to new data)



Machine Learning

Computer can learn without being explicitly programmed.

(and change behavior when exposed to new data)

- Vision, Natural Language Understanding, translation, sound
- Weather
- Finance, business, economics
- Science: DNA, Astronomy, Physics, Biology
- Strategy, from trading to games to wars

Why all this fuzz?

How we got here?

Hype

- First, there was Statistics (Econometrics, etc.)
- The focus was on CAUSALITY (correlation doesn’t imply causation -> the beer example)
- But then...
  - Power availability
  - Data availability
  - Who cares about causality: I want to **PREDICT**!

- **And it works :O**

Let’s dive deeper

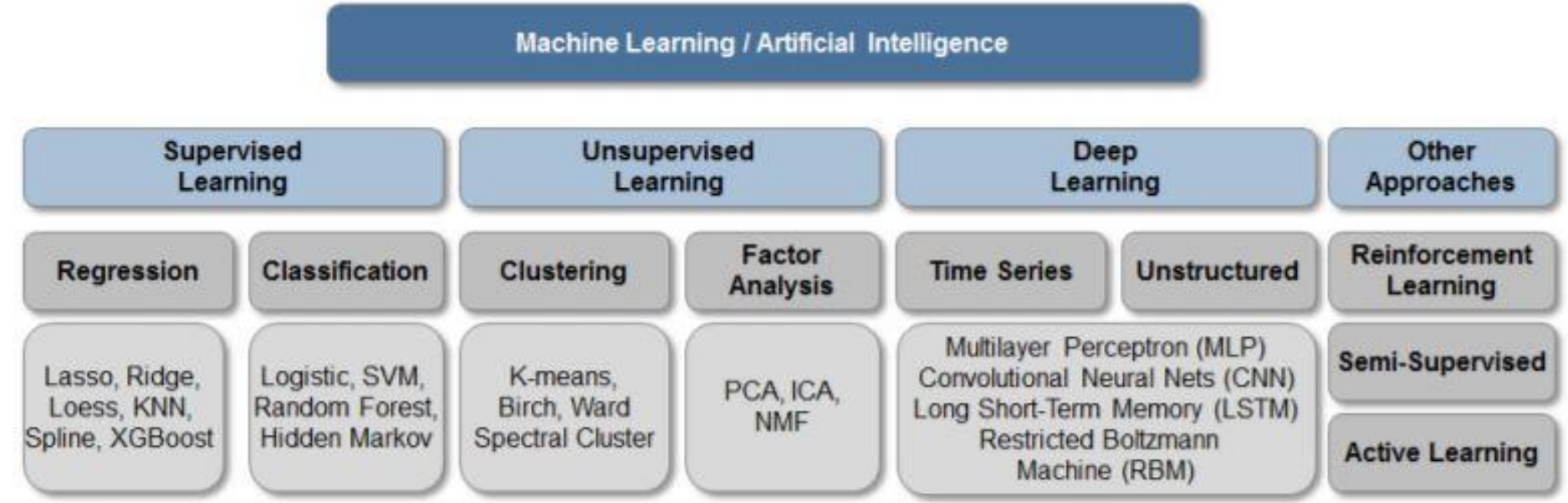
Cuz all these formulas are getting boring

Taxonomy: let's call stuff using proper names

- Supervised learning
  - Regression: a continuous target -> prices, attendances, time, etc
  - Classification: a discrete target -> categories, labels, 0/1
- Unsupervised learning
  - Clustering
  - Factor analysis
- Deep Learning
- Reinforcement learning, Recommendations systems, ...

J.P. Morgan is fancy and everything

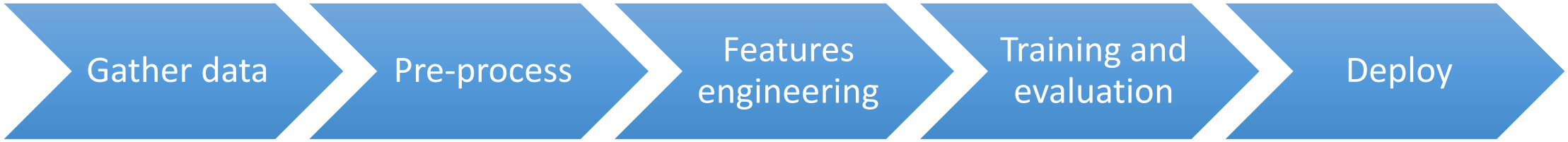
Figure 39: Classification of Machine Learning techniques



Source: J.P.Morgan Macro QDS

Pipeline

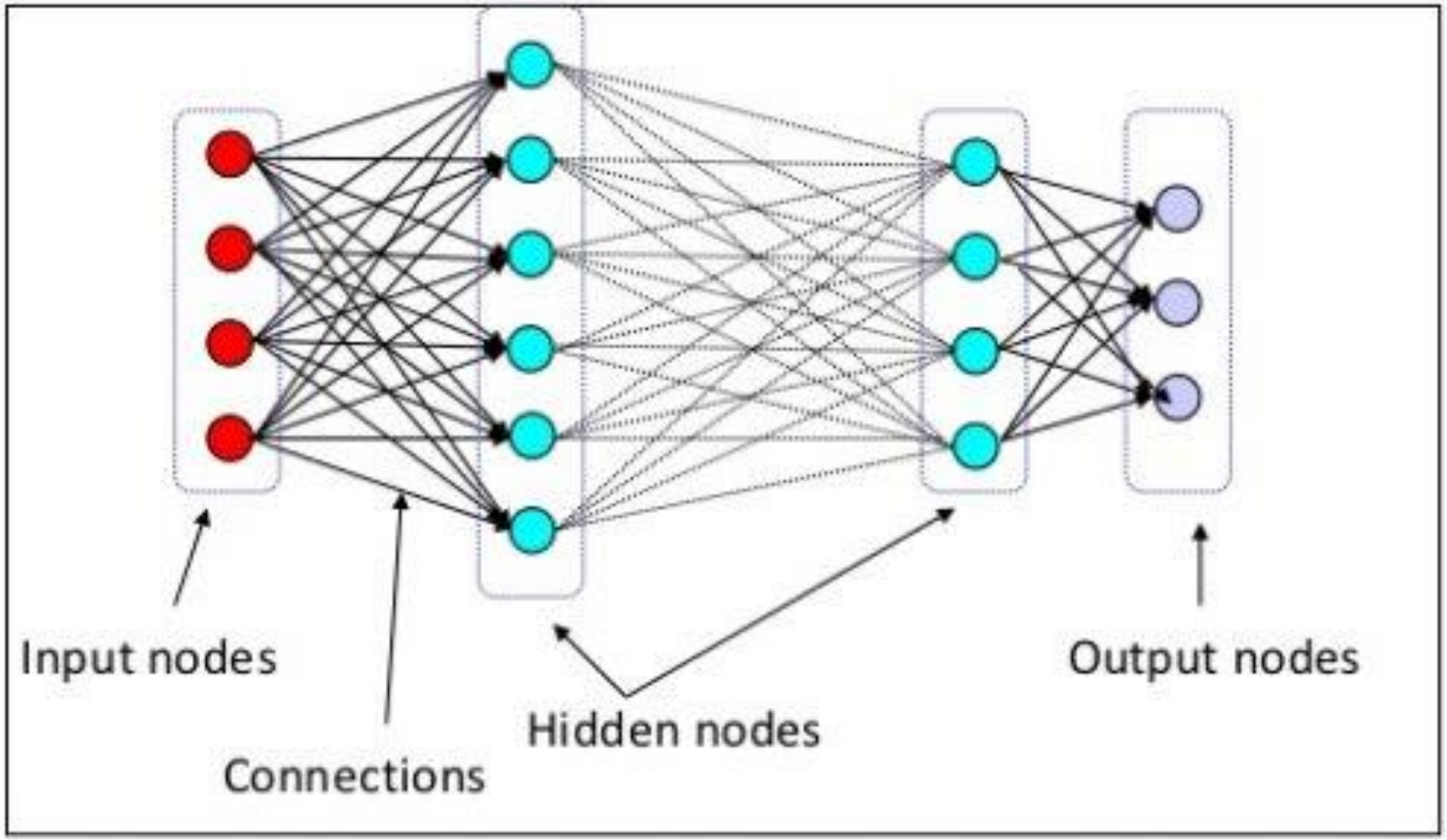
Workflow


Gather data
Pre-process
Features engineering
Training and evaluation
Deploy

Deep Learning

When features are generated by the model

The Architecture

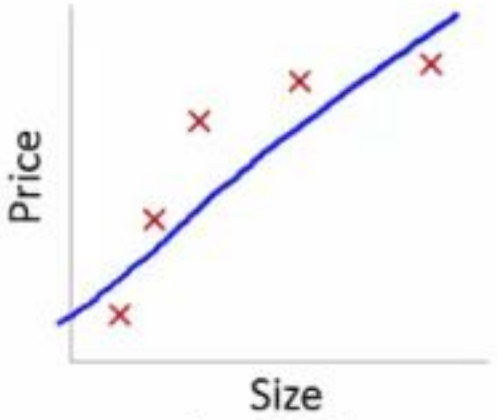


Generalization

ie. it needs to work outside your bedroom

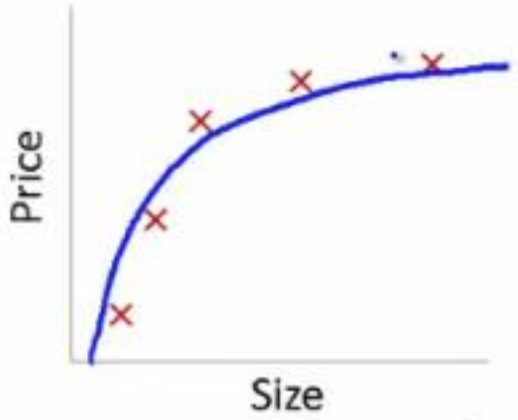
Not generalizing well

- Problem: the model is not representing the world “well enough”



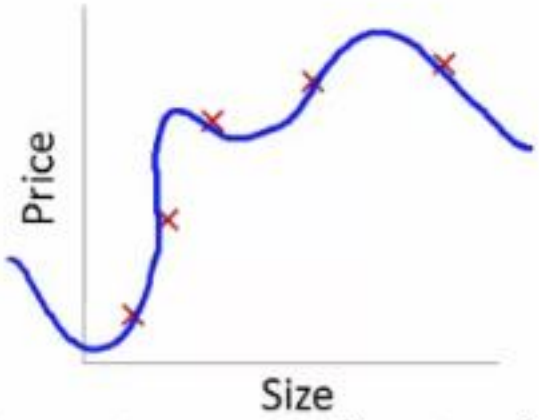
$\theta_0 + \theta_1 x$

High bias
(underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

“Just right”



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance
(overfit)

Split!

- How to identify & solve the generalization problem?
- The idea of the TRAINING set and the TEST set: split data!
- To solve it:
  - Over: Not over-complicating it, introducing regularization, not introducing too many features, etc.
  - Under: Introducing non-linearity, more data, more features

Data data data

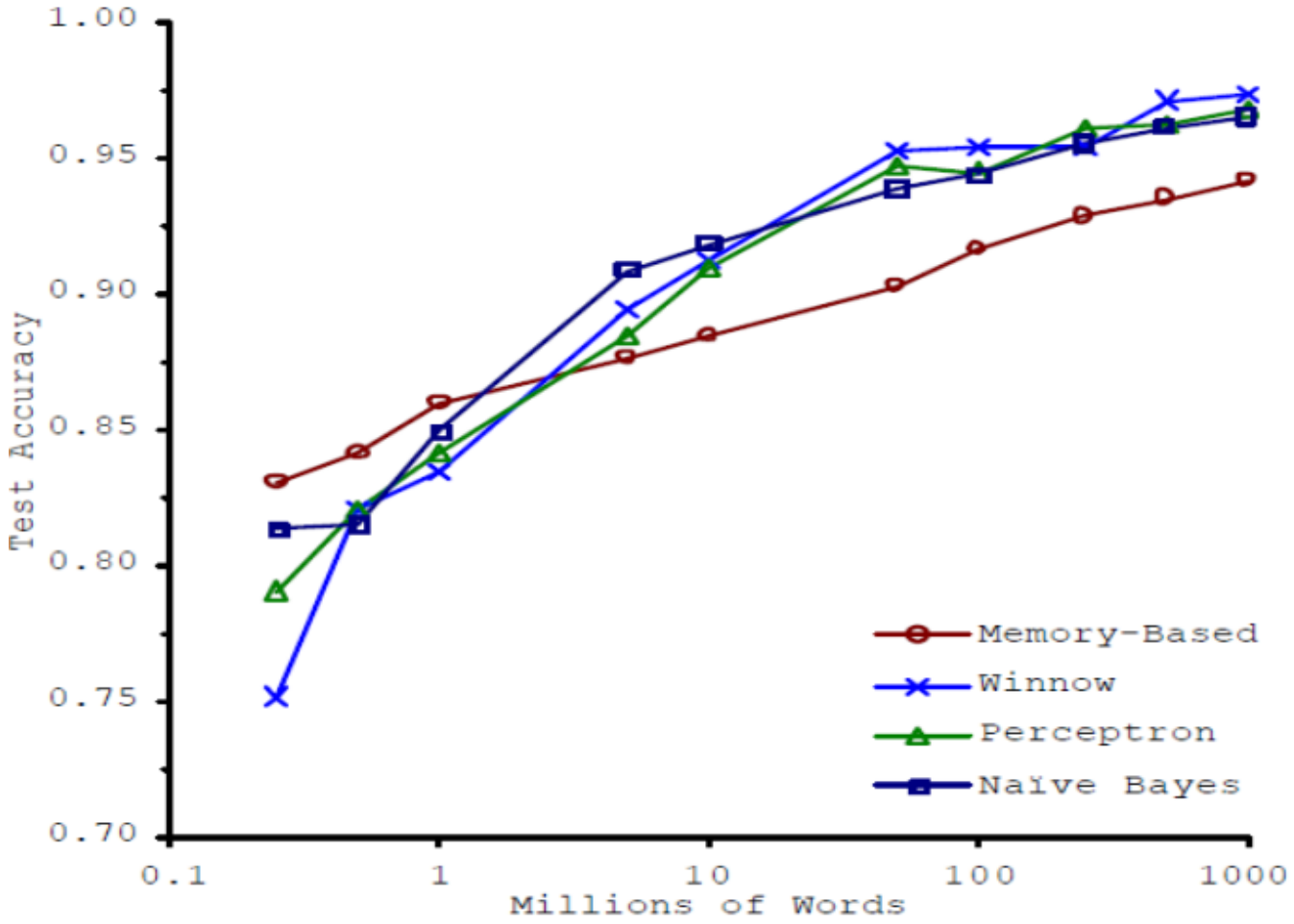
I’m gonna say this again: data data data

It all comes down to data

Google doesn’t buy satellite images just because it likes space. And yet it open sourced Tensor Flow.

Guess why?

“It’s the data, silly!”



Ok, nice. Can I see some code?

Okidoki – let’s go for an Hello World

Libraries

- Python:
  - Tensorflow
  - Keras
  - scikit-learn
  - ...
- R
- Matlab
- C++, Java, etc

Hello World (ie. a classification problem)

```
clf = LogisticRegression()
clf.fit(X_train, y_train)
clf.predict(X_test)
```

Real world case

```
>>> df.iloc[0]
id                                                          1924260491
username                                                joseph_scinto_artist
full_name                                                   Joseph Scinto
full_name_emoticons_ratio                                               0
biography                          I am an artist from Long Island, NY. I work in...
biography_length                                                      148
biography_new_lines                                                     0
biography_words_count                                                  28
biography_unique_words_ratio                                     0.964286
biography_emoticons_ratio                                      0.00675676
biography_special_chars_ratio                                    0.222973
external_url                                                            0
followed_by                                                           504
follows                                                              1080
following_followers_ratio                                         2.14286
is_verified                                                             0
profile_pic_url                    https://scontent-atl3-1.cdninstagram.com/t51.2...
connected_fb_page                                                       0
pictures_count                                                        445
pictures_average_size_ratio                                      0.0336407
pictures_average_size_abs                                            25830
```

Real world case

```
>>> df_u.shape
(39942, 53)
```

a skewed classification problem (90-10 class ratio)

Real world case

```
km = KMeans(n_clusters=3, init='k-means++’)
df_u['clusters'] = km.fit_predict(X)
df_u = pd.get_dummies(df_u, columns=['clusters'])


df_u = df.drop(['clusters', 'username', 'full_name',
                'biography', 'profile_pic_url'],
axis=1)
```

Real world case

```
X = df_u.drop(['y'], axis=1).values
y = df_u.y.values


X_t, X_test, y_t, y_test = train_test_split(X, y,
test_size=0.3)


scaler = preprocessing.StandardScaler().fit(X_train)

X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

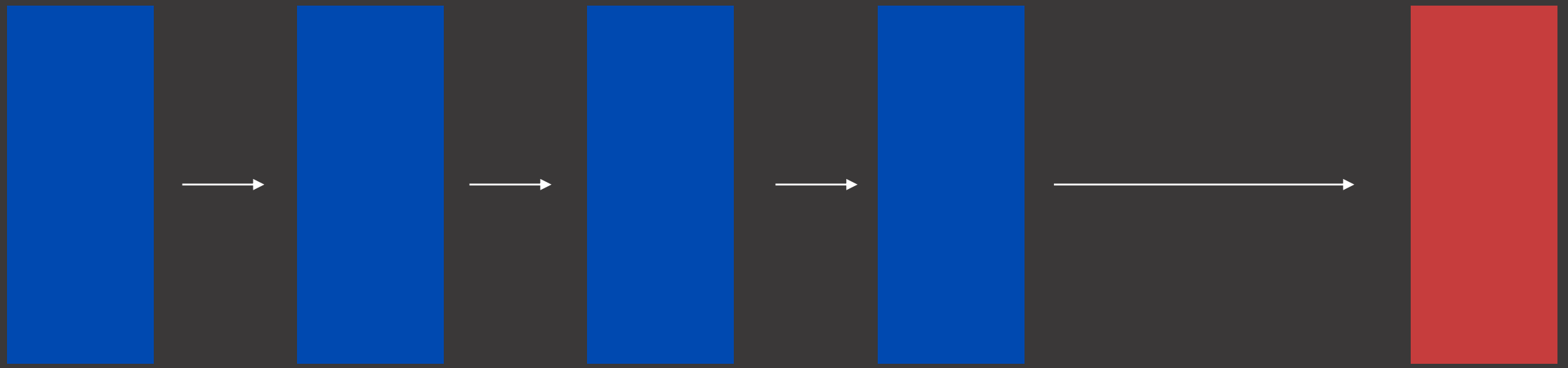
Real world case

```
array([[-0.18360918,  1.51476032, -0.56034339, ..., -0.04451727,
        -0.08492363, -0.05780098],
       [-0.18360918,  0.91050216,  2.46210571, ..., -0.04451727,
        -0.08492363, -0.05780098],
       [-0.18360918, -0.77009084, -0.56034339, ..., -0.04451727,
        -0.08492363, -0.05780098],
       ...,
       [-0.18360918,  0.28736093,  0.64863625, ..., -0.04451727,
        -0.08492363, -0.05780098],
       [-0.18360918, -1.27993367, -0.56034339, ..., -0.04451727,
        -0.08492363, -0.05780098],
       [-0.18360918, -0.24136495, -0.56034339, ..., -0.04451727,
        -0.08492363, -0.05780098]])
```

Real world case

```
model = xgb.XGBClassifier(**{
    'objective': 'multi:softprob',
    'eval_metric': 'mlogloss',
    'num_class': 2,
    'scale_pos_weight': int(sum(y_t == 0) / sum(y_t == 1)),
    'min_child_weight': 5.396524752336326,
    'max_delta_step': 4.870563641238184,
    'gamma': 0.5656650289383957 6,
    'subsample': 0.9747334463017736 1,
    'colsample_bylevel': 0.5566469887212128 8,
    'colsample_bytree': 0.7731965601982300 2,
    'alpha': 3.766066440971474 4,
    'eta': 0.6800311616183003 5,
    'max_depth': int(7.400432192303608)}).fit(X_t, y_t)
```

Real world case



Raw data

Pre-processing
(normalization)

Clustering
(K-Means)

Classification
(XGBoost)

Output
(1 or 0)

Real world case

```
model_score(model, X_test, y_test)
```

```
accuracy:          0.925395405915714

precision:         0.796434418802985

recall:            0.3925205408534763 43

f1:       0.5240102854871613

roc_auc:           0.8843056884916368
```

Thanks :)

-> ping me if you want to know more about Uniwhere, our open data projects, etc. :)

Q&A!