TITANIC

Source (among others):
https://www.coursera.org/learn/competitive-data-science/home/welcome
https://www.coursera.org/learn/machine-learning/home/welcome
Voir liens inclus

MISCELLANEOUS

- Quand les données sont mises en forme, possibilité d'utiliser pickle:

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.read_pickle.html

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.to_pickle.html

train.to_pickle('train_dataframe.pkl')

train = pd.read_pickle('train_dataframe.pkl')

- Détruire les variables inutiles avant predict par exemple

del train, x1, x2, y1, y2

- Timing

start_time = time.time()

print('[{}] Start XGBoost Training'.format(time.time() - start_time))

print('[{}] Finish XGBoost Training'.format(time.time() - start_time))

FEATURE PREPROCESSING

Numeric

- Scaling (non-tree-based models):

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

X=(X-X.min())/(X.max()-X.min())

http://sebastianraschka.com/Articles/2014_about_feature_scaling.html

X=(X-X.mean())/X.std()

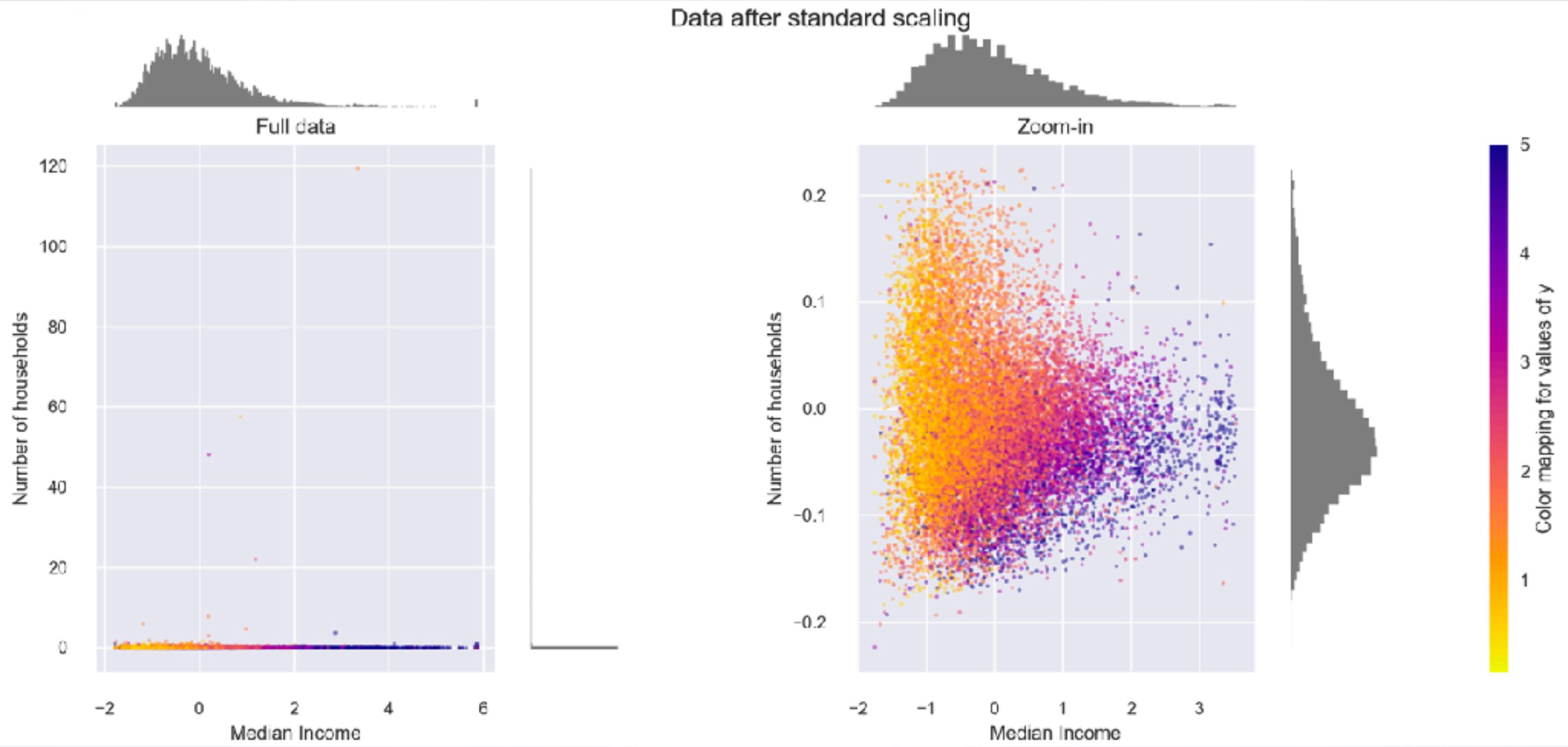
http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html (bug)

https://www.coursera.org/learn/machine-learning/lecture/xx3Da/gradient-descent-in-practice-i-feature-scaling

FEATURE PREPROCESSING

Numeric

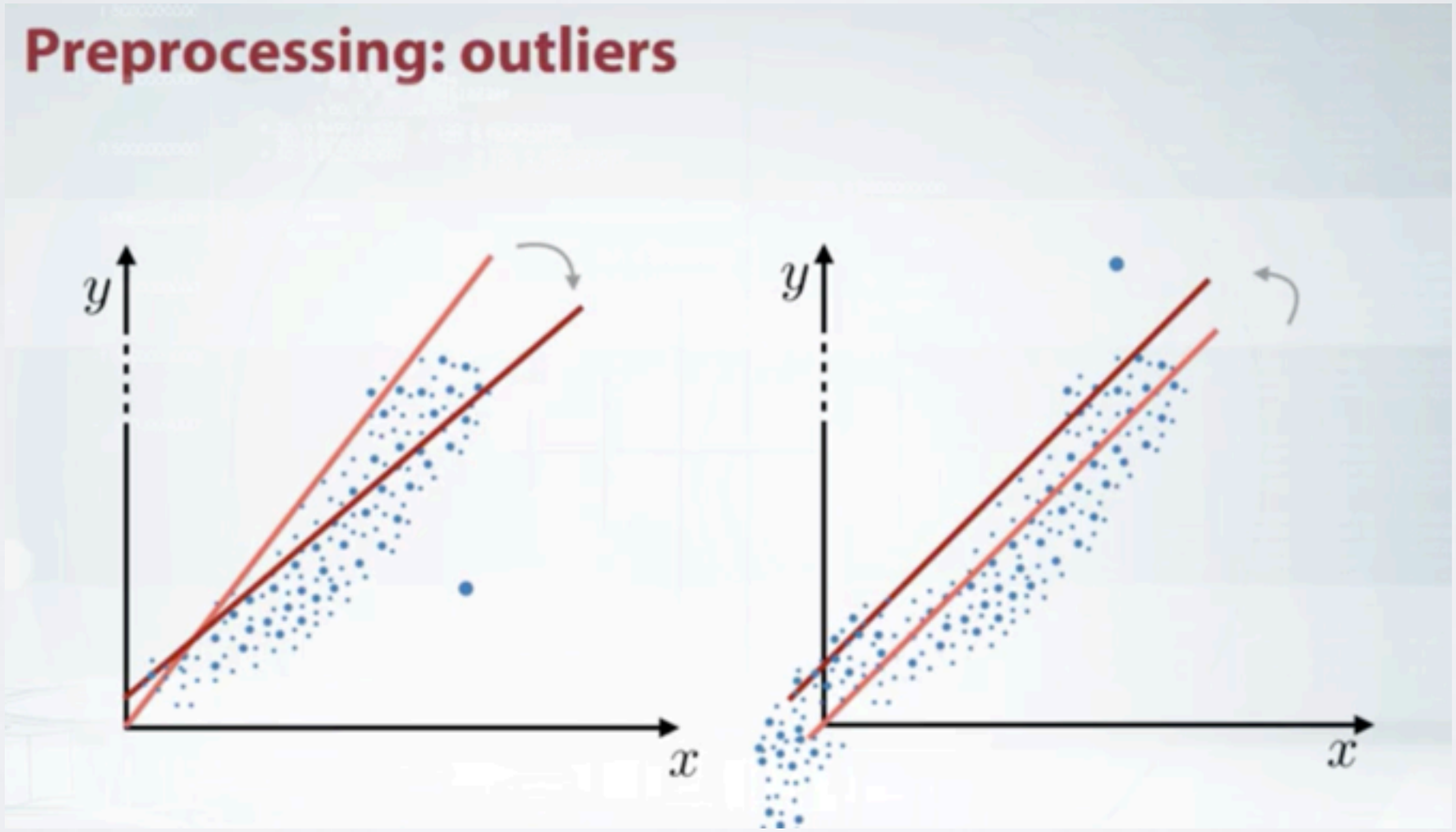
- Scaling :



FEATURE PREPROCESSING

Numeric

- Outliers: utilisation de np.percentile et np.clip

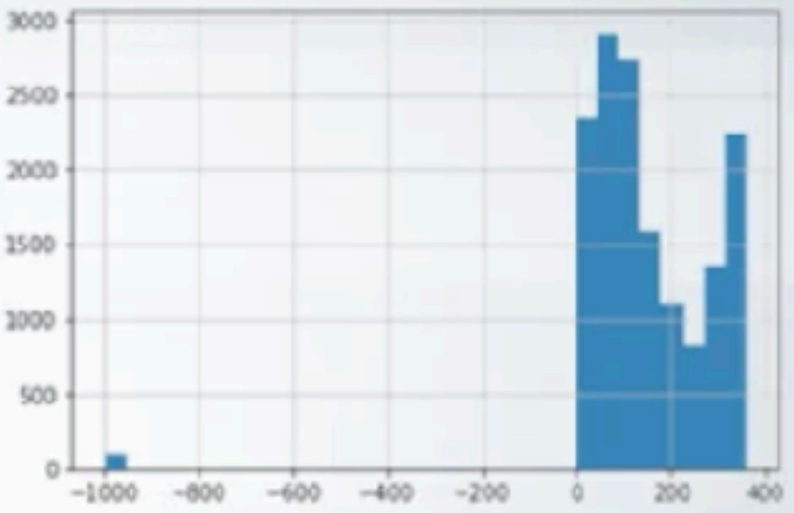


FEATURE PREPROCESSING

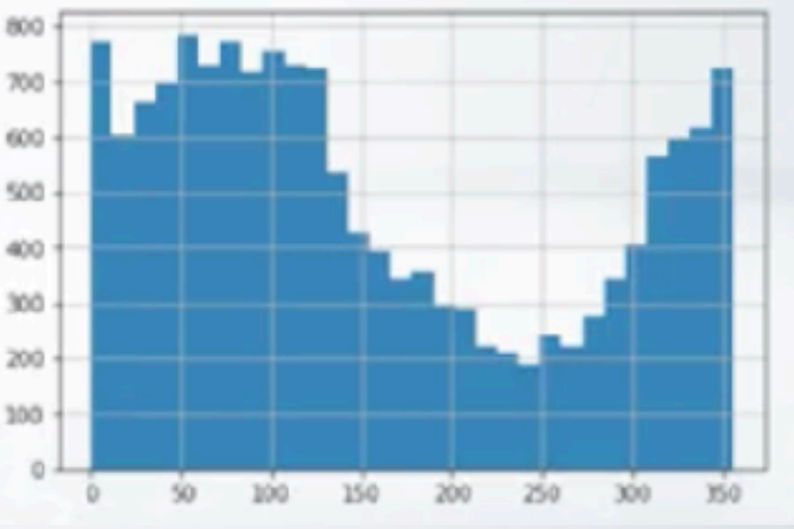
Numeric

- Outliers: utilisation de np.percentile et np.clip

```
In [17]: pd.Series(x).hist(bins=30) ;
```




```
In [18]: UPPERBOUND, LOWERBOUND = np.percentile(x, [1, 99])
y = np.clip(x, UPPERBOUND, LOWERBOUND)
pd.Series(y).hist(bins=30) ;
```



FEATURE PREPROCESSING

Numeric

- Rankdata

https://docs.scipy.org/doc/scipy-0.16.0/reference/generated/scipy.stats.rankdata.html

```
>>> from scipy.stats import rankdata
>>> rankdata([0, 2, 3, 2])
array([ 1. ,  2.5,  4. ,  2.5])
>>> rankdata([0, 2, 3, 2], method='min')
array([ 1.,  2.,  4.,  2.])
>>> rankdata([0, 2, 3, 2], method='max')
array([ 1.,  3.,  4.,  3.])
>>> rankdata([0, 2, 3, 2], method='dense')
array([ 1.,  2.,  3.,  2.])
>>> rankdata([0, 2, 3, 2], method='ordinal')
array([ 1.,  2.,  4.,  3.])
```

FEATURE PREPROCESSING

Numeric

- Transformation (neural networks): utilisation du log ou de la racine carrée

Voir cours coursera sur Gaussian Distribution

np.log(1+x), np.sqrt(x+2/3), etc.

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html

FEATURE PREPROCESSING

Categorical

- Transformation d'une colonne (comme pclass) en plusieurs colonnes avec 0 et 1 (linear method, kNN, neural net)

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.get_dummies.html

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

One-hot encoding

pclass
1
2
1
3

pclass==1	pclass==2	pclass==3
1		
	1	
1		
		1

FEATURE PREPROCESSING

Ordinal

- Attention a la différence entre numérique et ordinale: exemple: la classe dans le Titanic est ordinale.

Label encoding, frequency encoding:

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.factorize.html

K
embarked
S
C
S
S
S
Q
S
S

1. Alphabetical (sorted)
   [S,C,Q] -> [2, 1, 3]

   `sklearn.preprocessing.LabelEncoder`

2. Order of appearance
   [S,C,Q] -> [1, 2, 3]

   `Pandas.factorize`

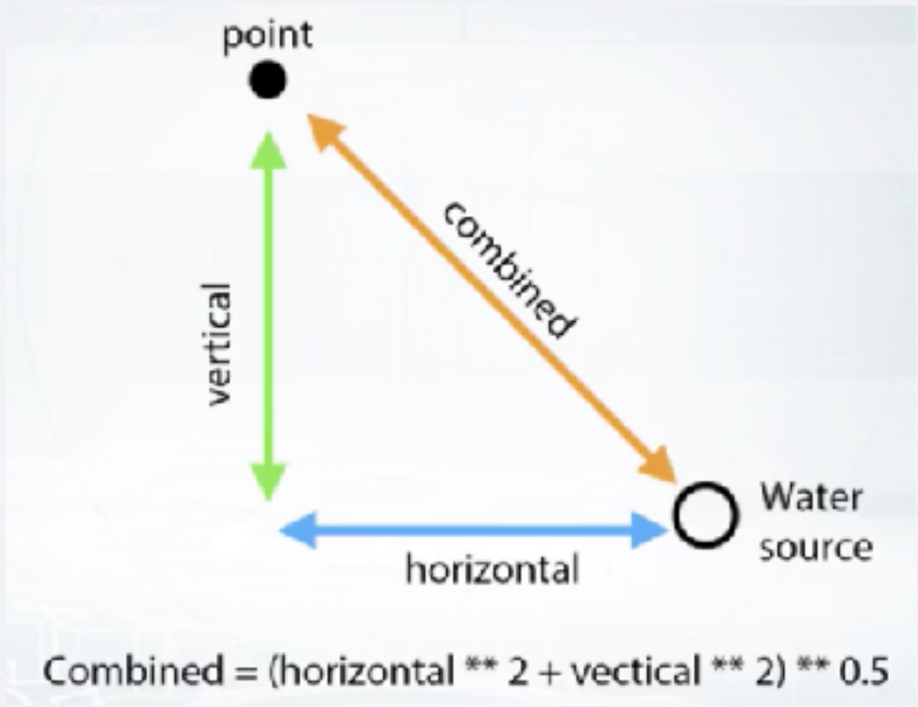
```
encoding = titanic.groupby('Embarked').size()
encoding = encoding/len(titanic)
titanic['enc'] = titanic.Embarked.map(encoding)
```

FEATURE GENERATION

- Création de nouvelles features: (Exemple: création de la feature prix par metre carré)

https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/

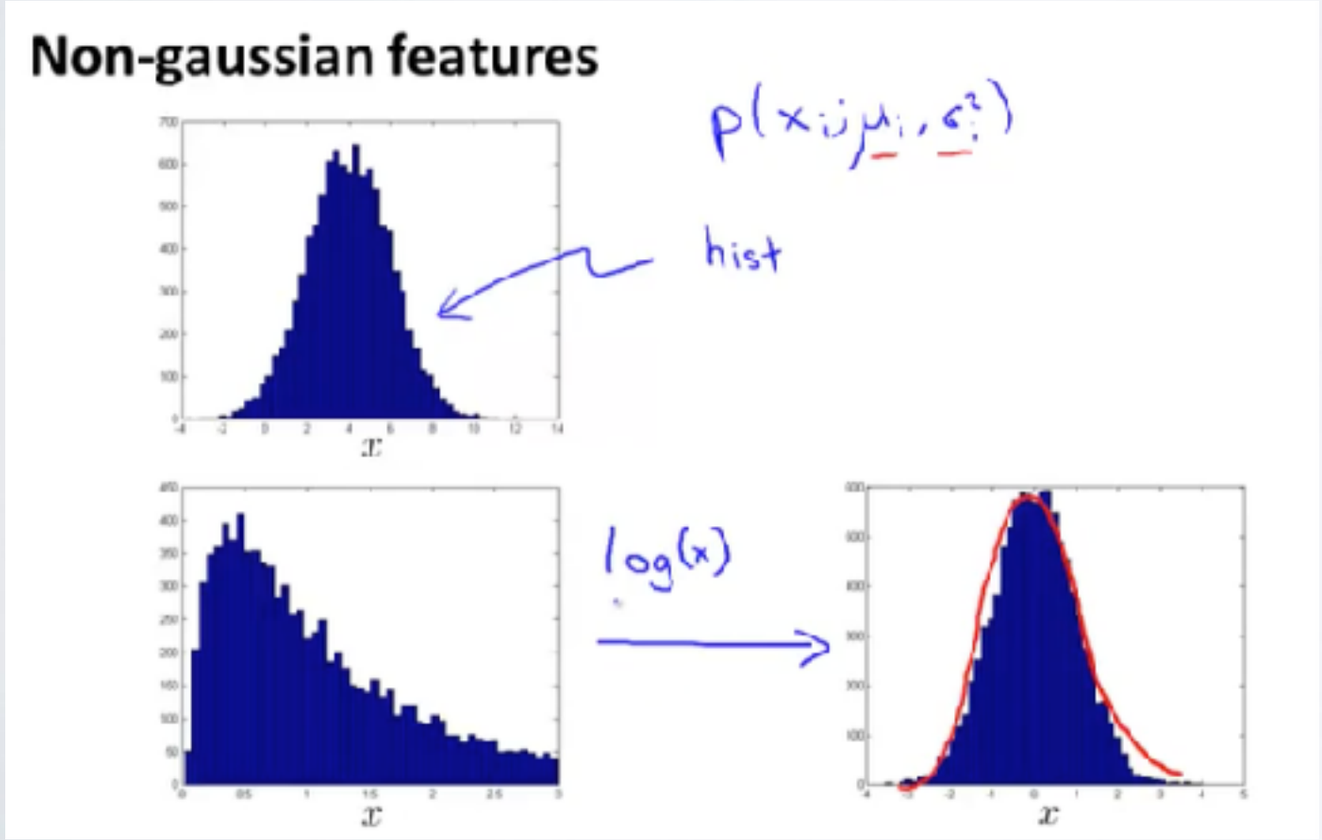
https://www.quora.com/What-are-some-best-practices-in-Feature-Engineering

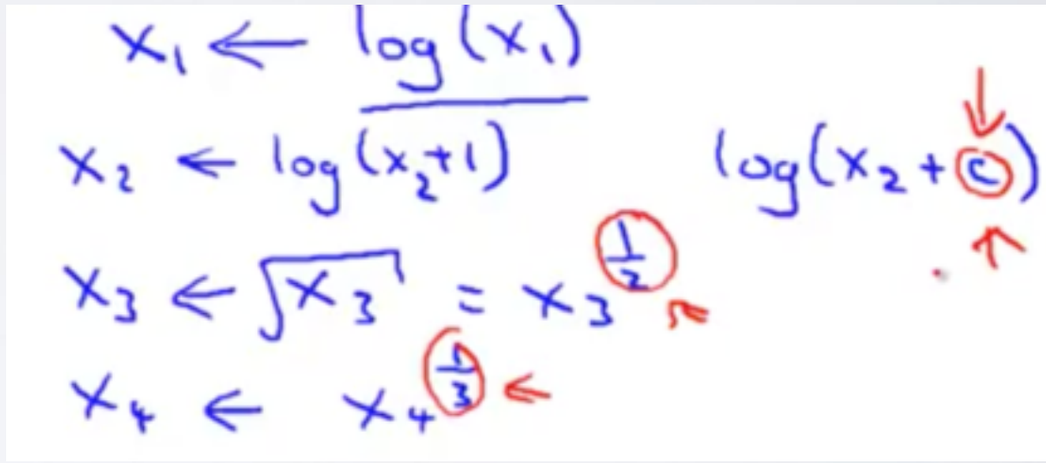


FEATURE GENERATION

Video à voir éventuellement:

https://www.coursera.org/learn/machine-learning/lecture/LSpXm/choosing-what-features-to-use





FEATURE GENERATION

https://www.coursera.org/learn/machine-learning/lecture/LSpXm/choosing-what-features-to-use

Monitoring computers in a data center

Choose features that might take on unusually large or small values in the event of an anomaly.

- x_1 = memory use of computer
- x_2 = number of disk accesses/sec
- x_3 = CPU load
- x_4 = network traffic

$$x_5 = \frac{\text{CPU load}}{\text{network traffic}} \qquad x_6 = \frac{(\text{CPU load})^2}{\text{network traffic}}$$