Brief ML Overview and NLP Background

Outline

1. Brief Overview of Machine Learning
2. NLP Basics

Appendix: Project Discussion

Brief Overview of Machine Learning

Machine Learning | Basic Definitions

Model A series of steps used to calculate a prediction from some input data.

Feature An individual measurable property or characteristic of a phenomenon.

Target The feature of a dataset about which you want to gain a deeper understanding.

Algorithm A set of steps used to build a model.

Metric A score used to evaluate the accuracy of a model. There are many different options.

Loss Function A function or calculation to minimize for model training. There are many different options.

Classification A type of supervised learning problem in which the target can only take one unique value/observation.

Regression A type of supervised learning problem in which the target is a number like time or dollars.

Multilabel A type of supervised learning problem in which the target can take multiple labels.

Preprocessing A series of steps used to prepare data for an algorithm in order to build a model.

Supervised Learning A type of machine learning in which labeled data points are used to build a model, whose primary use is [usually] making predictions.

Machine Learning A widely used and very ambiguous term that refers to the art of building models.

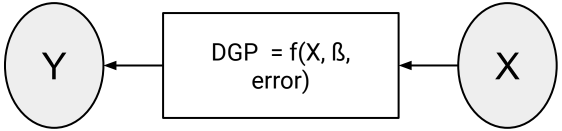
Deep Learning A type of machine learning using a neural network architecture with 1 or more (usually more hence “deep”) hidden layers.

Machine Learning - Shifting Paradigms

In, 1959 Arthur Samuel defined machine learning as a “Field of study that gives computers the ability to learn without being explicitly programmed”

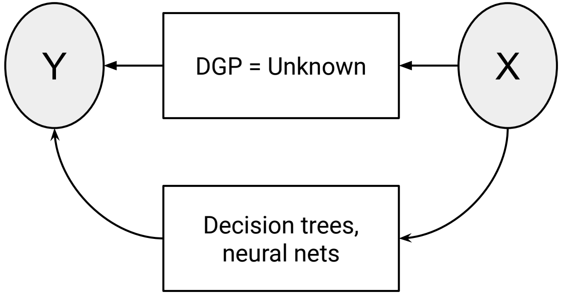
Traditional Models

- Often focused on parameters
- Expertly curated datasets
- Strong assumptions about the data generating process (DGP)
- Goodness of fit, p-values, residual analysis



Machine Learning Models

- Empirically driven
- Large datasets (wide and long)
- Weak assumptions about the data
- Accuracy and generalizability
- More focus on prediction vs inference



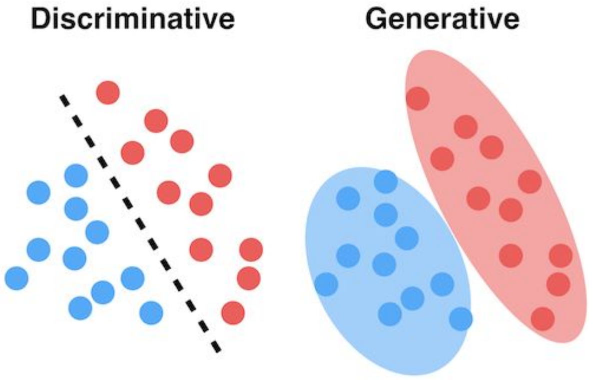
Terminology Translation

Statistics	**Machine Learning**
Data point, record, row	Example, instance
Response variable, dependent variable, endogenous variable, regressand	Label, output, target
Variable, covariate, predictor, independent variable, exogenous variable, regressor	Feature, input
Regression	Supervised Learning, regression

https://insights.sei.cmu.edu/sei_blog/2018/11/translating-between-statistics-and-machine-learning.html

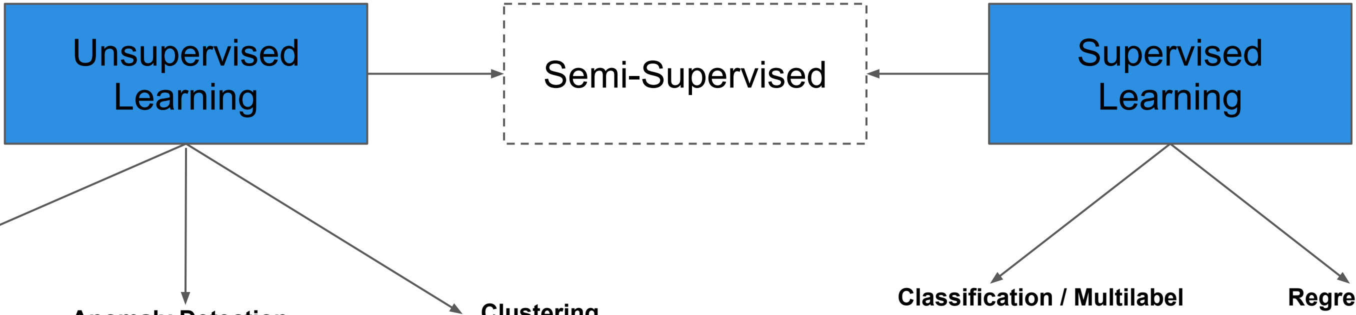
Frameworks

- Supervised, unsupervised, semi-supervised, self-supervised
- Reinforcement Learning
- Generative vs. discriminative models
- Representation Learning
- Parametric vs non-parametric



Not to be confused with GenAI

Example Types of Machine Learning



Unsupervised Learning

Semi-Supervised

Supervised Learning

Dimensionality Reduction

- Big Data Visualization
- Structure Discovery
- Text Summarization

Anomaly Detection

- Anti-Money Laundering
- Price Insight
- Weather Patterns

Clustering

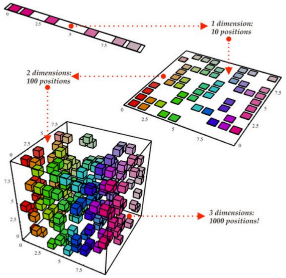
- Segmentation
- Image Segmentation
- Customer Retention

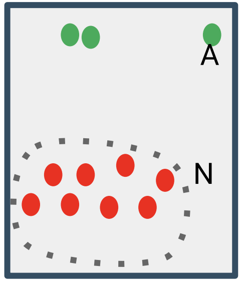
Classification / Multilabel

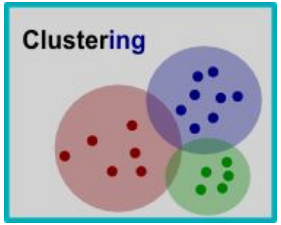
- Fraud Detection
- Image Classification
- Customer Retention
- Diagnostics
- Topic Modeling

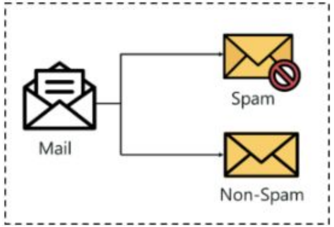
Regression

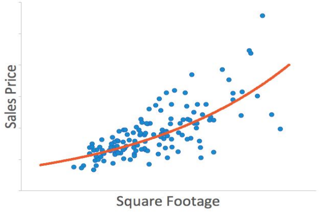
- Temperature Forecasting
- Market Forecasting
- Life Expectancy Forecasting
- House Price Prediction
- Prepayment speed


1 dimension: 10 positions
2 dimensions: 100 positions
3 dimensions: 1000 positions!





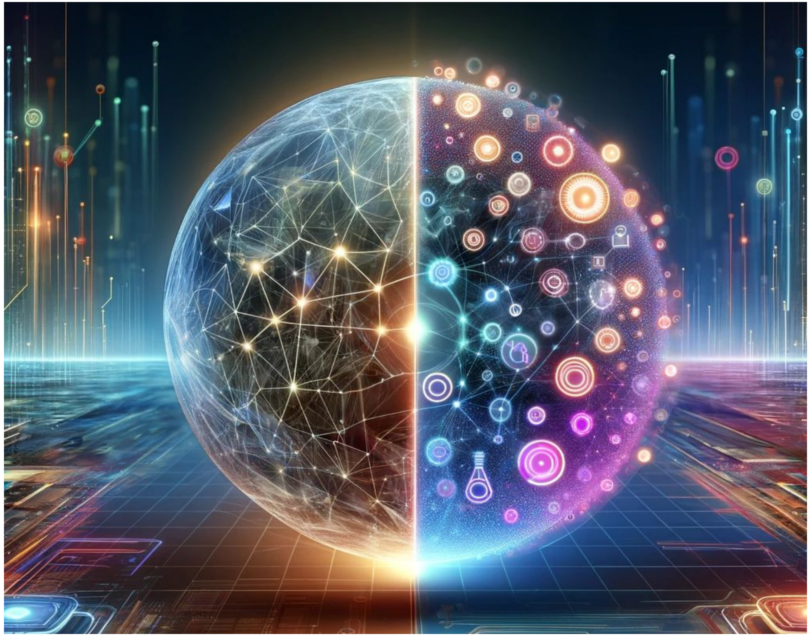




Buzzwords: Generative AI vs Predictive AI

Generative AI

- Still Machine Learning
- Typically generating strings or images
- Large Language Models (LLMs)
- DALLE, Mid-Journey, etc.
- Often trained in a self-supervised way
- **Example**: Answer questions like with Chat-GPT
- Can be thought of as Predictive AI, but purpose is for generation

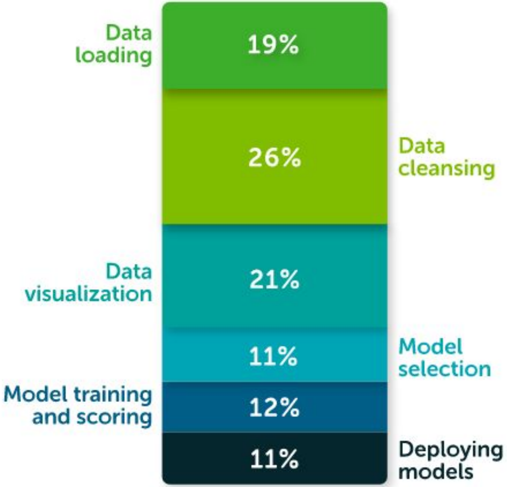


Predictive AI

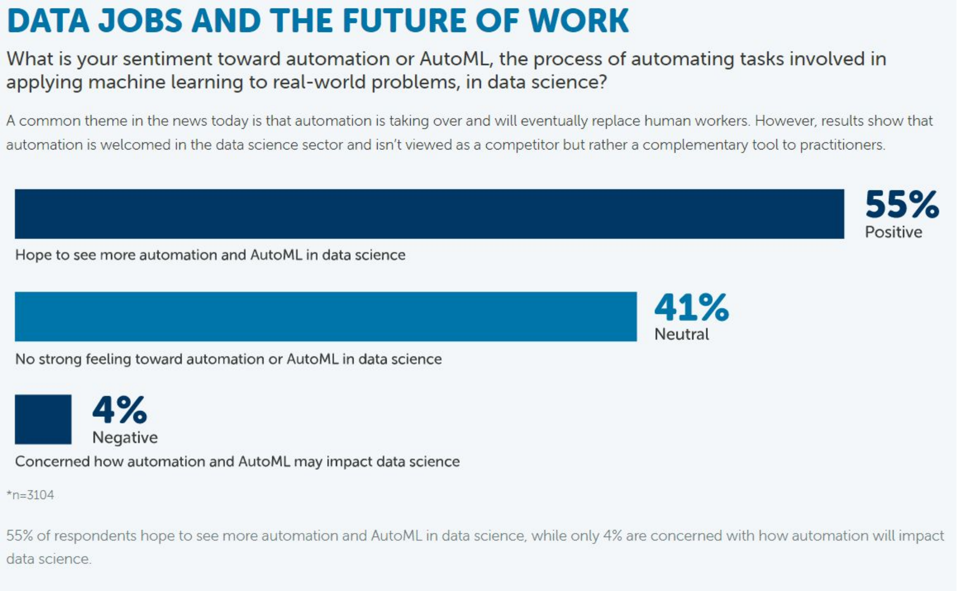
- Still Machine Learning
- Predict outcomes such as binary targets, probabilities, multi-class, multi-label, and continuous numeric data (regression)
- Typically trained with supervised learning
- **Example**: determine sentiment of text

It’s not just about modeling, but this is what we will focus on

How much of your time is spent on each of the following tasks?



https://www.anaconda.com/state-of-data-science-2020



DATA JOBS AND THE FUTURE OF WORK

What is your sentiment toward automation or AutoML, the process of automating tasks involved in applying machine learning to real-world problems, in data science?

A common theme in the news today is that automation is taking over and will eventually replace human workers. However, results show that automation is welcomed in the data science sector and isn’t viewed as a competitor but rather a complementary tool to practitioners.

55% Positive

Hope to see more automation and AutoML in data science

41% Neutral

No strong feeling toward automation or AutoML in data science

4% Negative

Concerned how automation and AutoML may impact data science

*n=3104

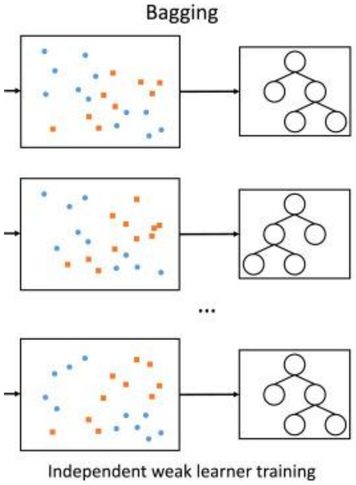
55% of respondents hope to see more automation and AutoML in data science, while only 4% are concerned with how automation will impact data science.

https://www.anaconda.com/state-of-data-science-2021

Sample ML Techniques: Boosting & Bagging

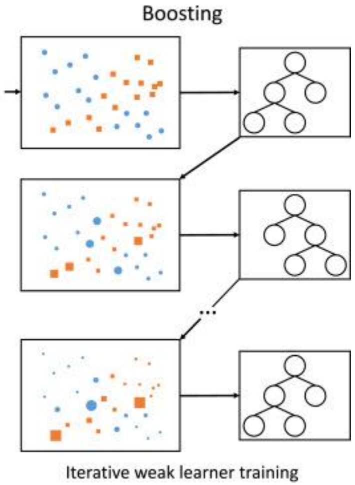
Bagging (e.g. Random Forest):

- Trees / models learn **independently and in parallel**
- A model is created on each subset
- Regularization and subsetting both observations and features are used to avoid overfitting (no learning rate)
- Final predictions are determined by combining predictions from all models
- Often helps reduces variance



Boosting (e.g. XGBoost):

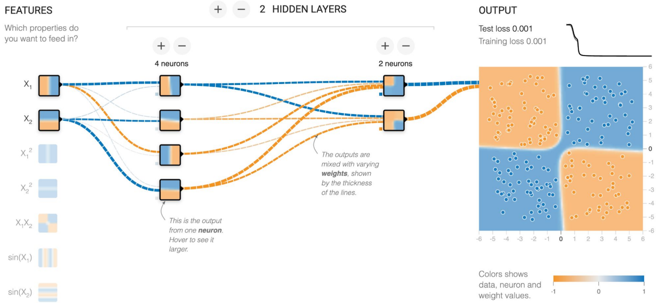
- Trees / models learn **sequentially and adaptively to improve model predictions**
- Learning rate, regularization, and subsetting both observations and features are used to avoid overfitting
- Generate predictions on the dataset and calculate errors
- Sequential models are created to correct errors from previous model(s)
- Often helps reduce bias and variance



Don’t forget about model stacking

Images from: https://www.sciencedirect.com/science/article/abs/pii/S1566253520303195

Deep Learning: Universal Approximation Theorem



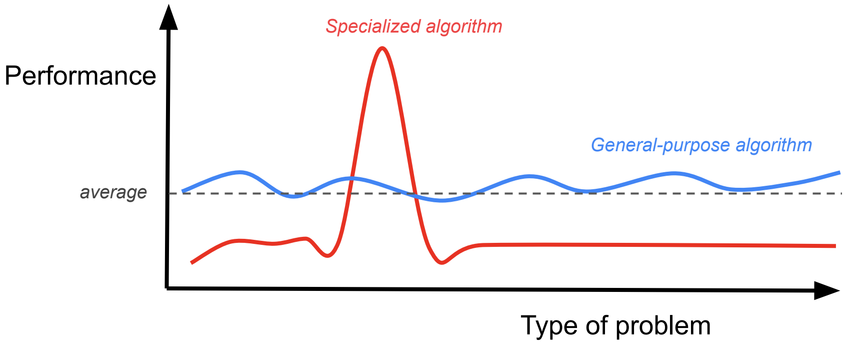
*This is the output from one **neuron**. Hover to see it larger.*

*The outputs are mixed with varying **weights**, shown by the thickness of the lines.*

Colors shows data, neuron and weight values.

“A neural network with one hidden layer containing a sufficient but finite number of neurons can approximate any continuous function to a reasonable accuracy, under certain conditions for activation functions”

So just use deep learning for everything, right? **No!**



There is No Free Lunch

No Free Lunch Theorem

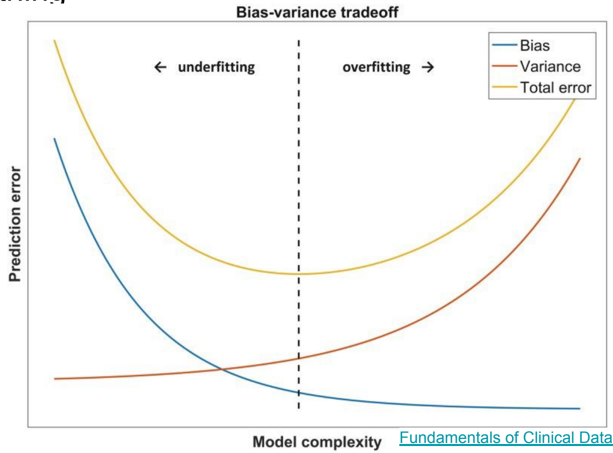
For any two learning algorithms $P_1(w|D)$ and $P_2(w|D)$ where

- w are the learnable parameters
- D is the training dataset (think of just the input here)
- P_i is the probability

I.e. No model rules them all so it is difficult to know what modeling approach to use in advance and some approaches may work better than others for your particular problem

independent of the number of training points, uniformly averaged over all target functions, F, $E_1(E|F,n) - E_2(E|F,n) = 0$ for the expected generalization error. This even hold true when fixing a training set D.

Related: Occam's razor or principle of parsimony: “One should not increase, beyond what is necessary, the number of entities required to explain anything”

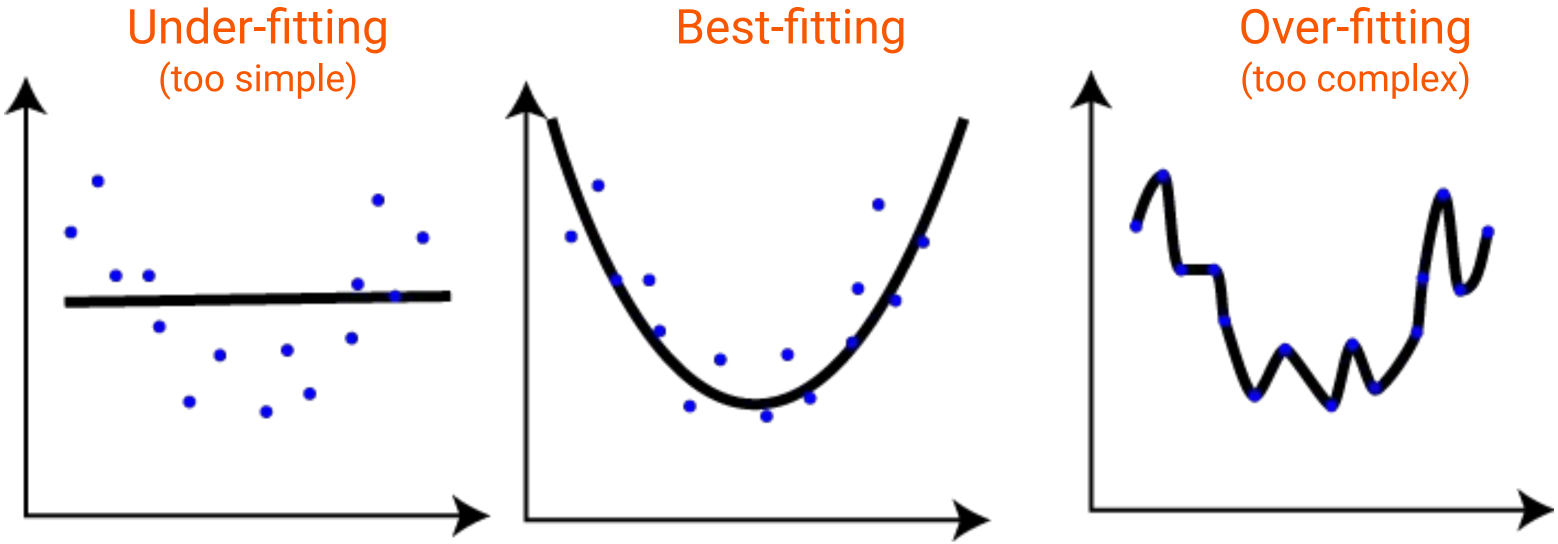


Fundamentals of Clinical Data Science, licensed under CC by 4.0.

Bias: error from wrong prior assumptions about the model

Variance: error from sensitivity to the training data

Over-fitting

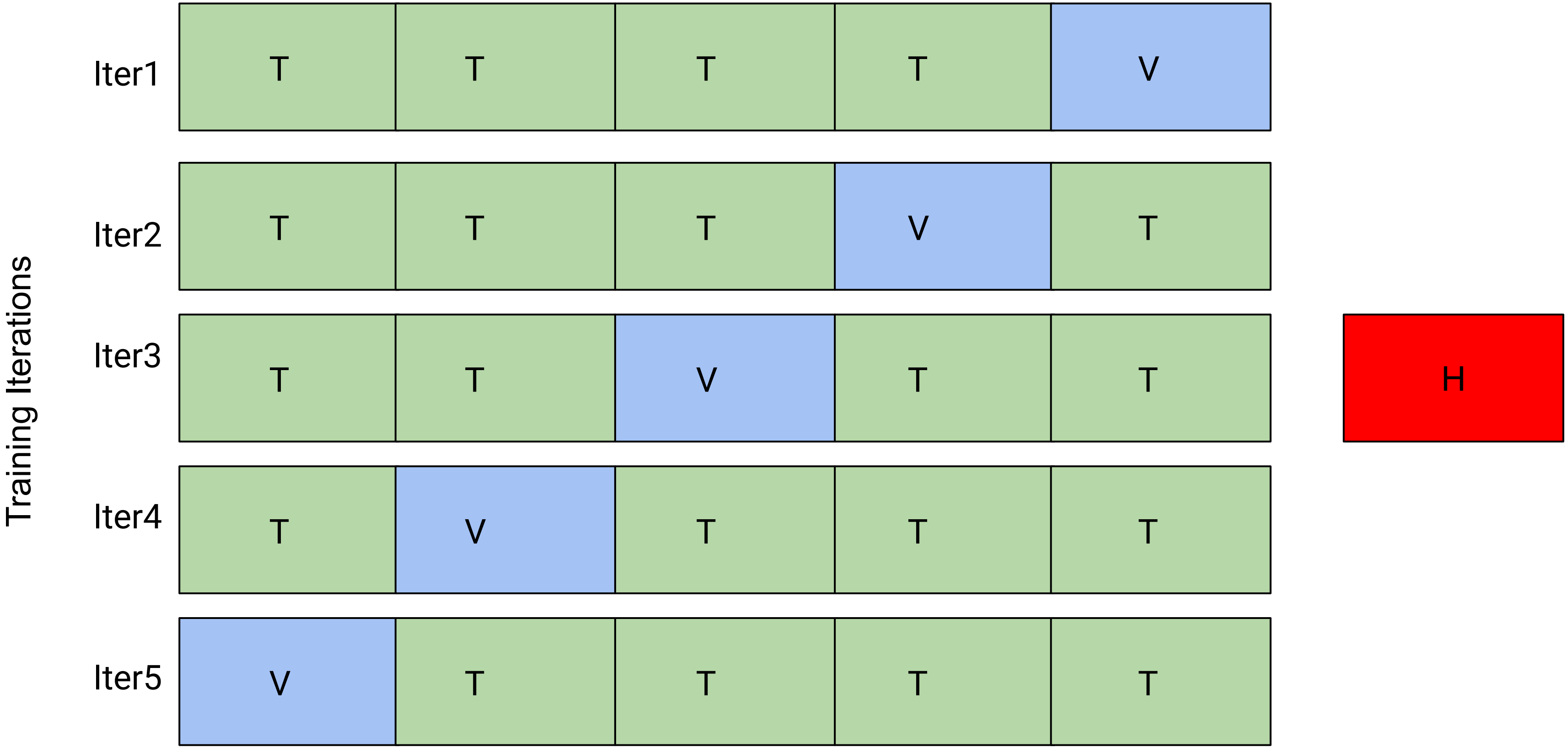


Under-fitting
(too simple)

Best-fitting

Over-fitting
(too complex)

k-Fold Cross Validation



Training Iterations

Iter1	T	T	T	T	V	
Iter2	T	T	T	V	T	
Iter3	T	T	V	T	T	H
Iter4	T	V	T	T	T	
Iter5	V	T	T	T	T	

Always use a holdout set if trying multiple experiments!

Data Partitioning

The main goal of training and validation is to understand how well our model generalizes to unseen data. More concretely, how we can expect our model to perform in the future? Below are examples of partitioning strategies and the following slides provide motivations for using them.

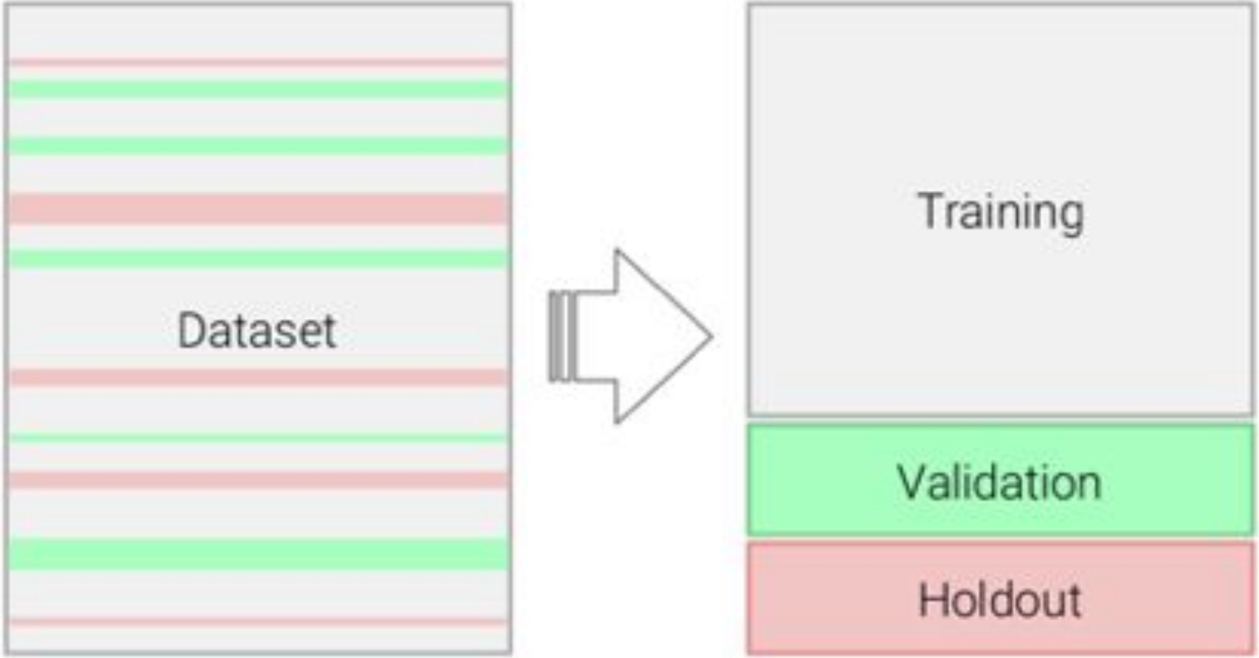
Partitioning Strategies

1) Random – a random assignment of records to partitions.
2) Group – Use grouping when you want all values from a particular group to be in the same partition. The user identifies a feature in the dataset used for grouping. Each partition may contain multiple values of the feature.
3) Stratified – Ratio or distribution preserving partitioning.
4) Out-of-time Validation / Backtesting – time based partitioning whereby training uses one period and validation uses another
5) Other / Custom - you may need an even more sophisticated approach to partition your data to appropriately measure out of sample performance

Random Sampling

Random Sampling is the most popular method for assigning observations to a partition. This method may be applied to any problem type.

When could this be a problematic approach to data partitioning?



Conventional machine learning approach (TVH): Random Sampling

Stratified Partitioning

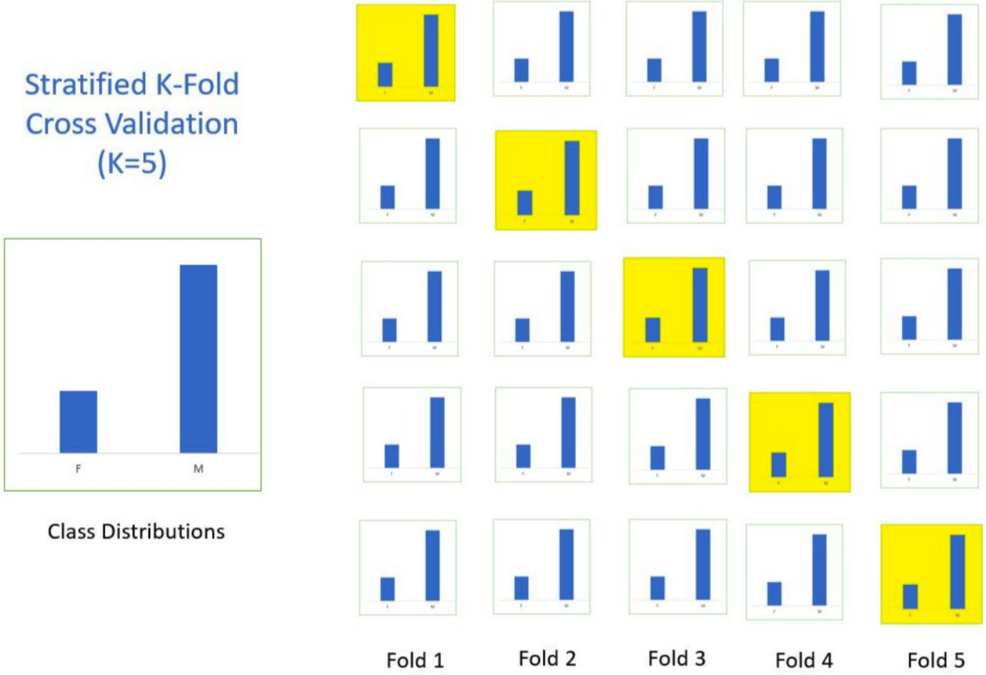
Stratified sampling is very useful for binary classification problems and regression problems following certain distributions.

In Binary Classification problems, more often than not, the variable of interest is unbalanced, meaning there are many more of one class than the other class (think rare or uncommon events).

The stratified partitioning strategy ensures that every partition preserves the same ratio of values for the prediction target as in the original data.

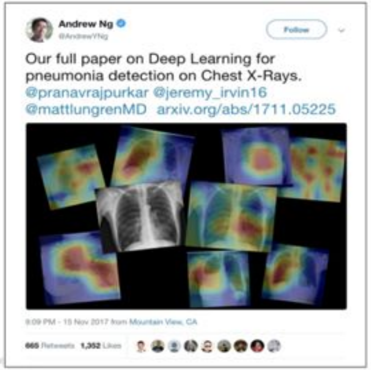
Thinking of using random sampling on unbalanced data? Worst case scenario your training data may not have any of the positive class, and will learning nothing.

How to do this for regression?



Group Partitioning

Grouping is a method used to prevent group leakage from affecting the validation results of your model. Group leakage can give you over confidence in your model. If you don't use grouping, your model may memorize attributes of the training data, and since the same group appears in training and validation, it performs exceptionally well on the validation dataset, but in general is useless. By grouping you learn only from all the observations in specific groups and predict on other groups.

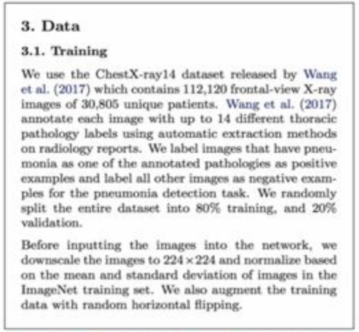


Andrew Ng
@AndrewYNg

Our full paper on Deep Learning for pneumonia detection on Chest X-Rays. @pranavrajpurkar @jeremy_irvin16 @mattlungrenMD arxiv.org/abs/1711.05225

9:09 PM - 15 Nov 2017 from Mountain View, CA

665 Retweets 1,352 Likes



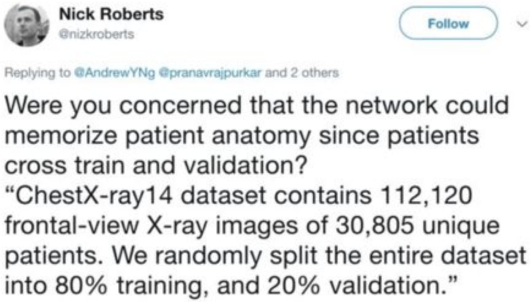
3. Data

3.1. Training

We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples for the pneumonia detection task. We randomly split the entire dataset into 80% training, and 20% validation.

Before inputting the images into the network, we downscale the images to 224×224 and normalize based on the mean and standard deviation of images in the ImageNet training set. We also augment the training data with random horizontal flipping.

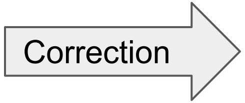
https://twitter.com/AndrewYNg/status/931026446717296640

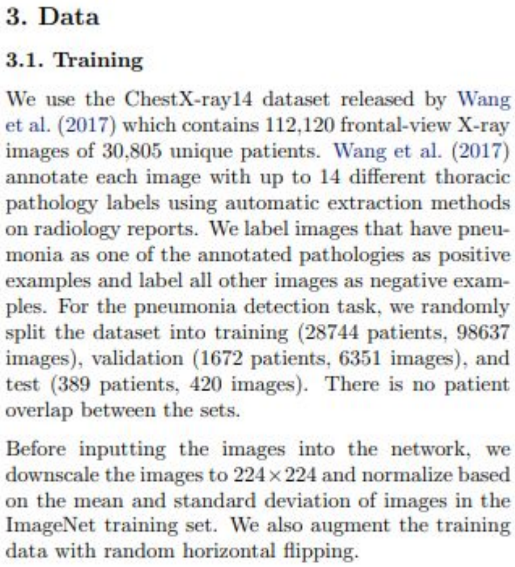


Nick Roberts
@nizkroberts

Replying to @AndrewYNg @pranavrajpurkar and 2 others

Were you concerned that the network could memorize patient anatomy since patients cross train and validation?
"ChestX-ray14 dataset contains 112,120 frontal-view X-ray images of 30,805 unique patients. We randomly split the entire dataset into 80% training, and 20% validation."





3. Data

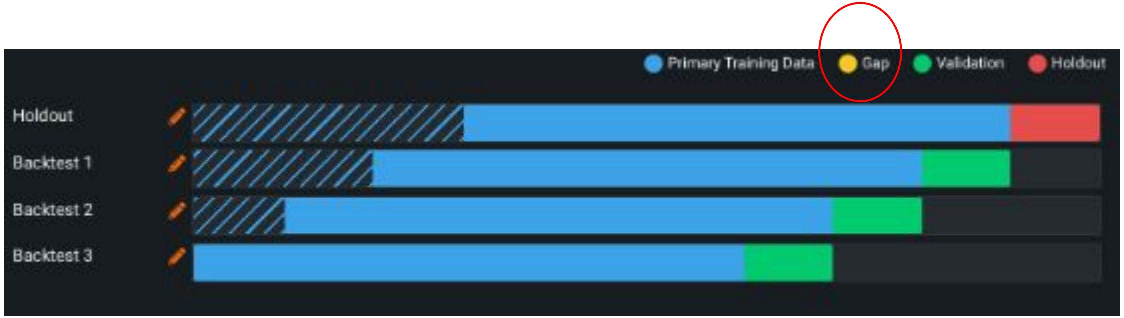
3.1. Training

We use the ChestX-ray14 dataset released by Wang et al. (2017) which contains 112,120 frontal-view X-ray images of 30,805 unique patients. Wang et al. (2017) annotate each image with up to 14 different thoracic pathology labels using automatic extraction methods on radiology reports. We label images that have pneumonia as one of the annotated pathologies as positive examples and label all other images as negative examples. For the pneumonia detection task, we randomly split the dataset into training (28744 patients, 98637 images), validation (1672 patients, 6351 images), and test (389 patients, 420 images). There is no patient overlap between the sets.

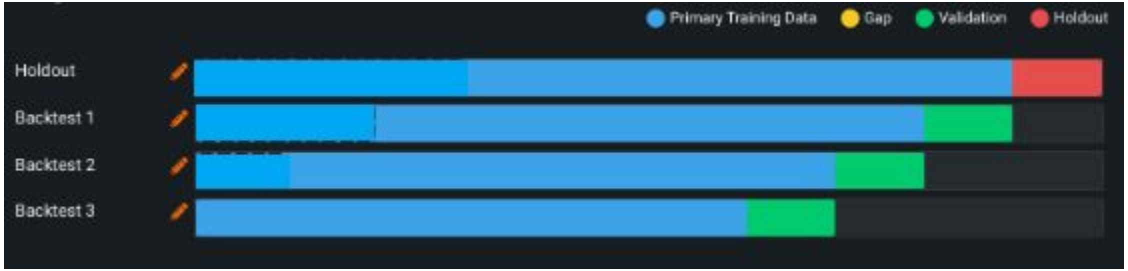
Before inputting the images into the network, we downscale the images to 224×224 and normalize based on the mean and standard deviation of images in the ImageNet training set. We also augment the training data with random horizontal flipping.

Out-of-time Validation / Backtesting

Time-based partitioning is essential when your data is not entirely independent and has any dependence on the time period. There are a number of acceptable ways to handle this, but random sampling is not appropriate and will overstate your model performance / understate the models generalization error.



https://docs.datarobot.com/en/docs/modeling/time/date-time.html



Note 1: Longer history is not always better. Think about your features and the dynamics you are modeling.
Note 2: You can create non-overlapping backtests as well

Model Interpretability

Importance:

Univariate methods:
- Correlation
- Alternating Conditional Expectation
- Mutual Information
- Cramer’s V

Model-based methods:
- “Leave it out”
- Permutation-based
- Tree-based / Information Gain

… many more

Sensitivity / Dependence:

- Partial Dependence Plots
- LIME
- ICE
- … many more

Explainability:

- SHAP
- Activation Maps
- Integrated Gradients
- … many more

Example Distances

Distance	Equation
Euclidean	$d(x, y) = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i \right)^2}$
Manhattan	$d(x, y) = \sum_{i=1}^{n} \lvert x_i - y_i \rvert$
Minkowski	$d(x, y) = \left(\sum_{i=1}^{n} \lvert x_i - y_i \rvert^p \right)^{\frac{1}{p}}$
Jaccard	$J(A, B) = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert}$
Cosine	$1 - \frac{x \cdot y}{\lVert x \rVert \cdot \lVert y \rVert}$

NLP Basics

Introduction to working with Text Data

The feature space

Number of Words in Oxford English Dictionary: ~210,000

Number of total English words: ~1,000,000

Number of Words in all the works of Shakespeare: ~24,000

Number of words the average English speaker knows: ~20,000-40,000

Text mining processes

- **Text preprocessing**
  - **Syntactic/Semantic text analysis**
  - **Regex, stemming, stop words, lemmatization, correcting spelling, etc.**
- **Features Generation**
  - **Bag of words / ngrams**
  - **Word2Vec / Fasttext**
  - **Deep Learning Models (e.g. transformers)**
- **Text/Data Mining**
  - **Supervised learning / Classification**
  - **Clustering- Unsupervised learning**
- **Mapping/Visualization Result interpretation**

Bag of Words

#	Text from Surveys
1	I’m free to go to the conferences
2	This conference is the best ever
3	This breakfast is the best ever
4	I only came to the conference for the free breakfast

Becomes

#	Stem and Remove Stop Words
1	~~I’m~~ free ~~to go to the~~ conference~~s~~
2	~~This~~ conference ~~is the~~ best ever
3	~~This~~ breakfast ~~is the~~ best ever
4	~~I~~ only came ~~to the~~ conference ~~for the~~ free breakfast

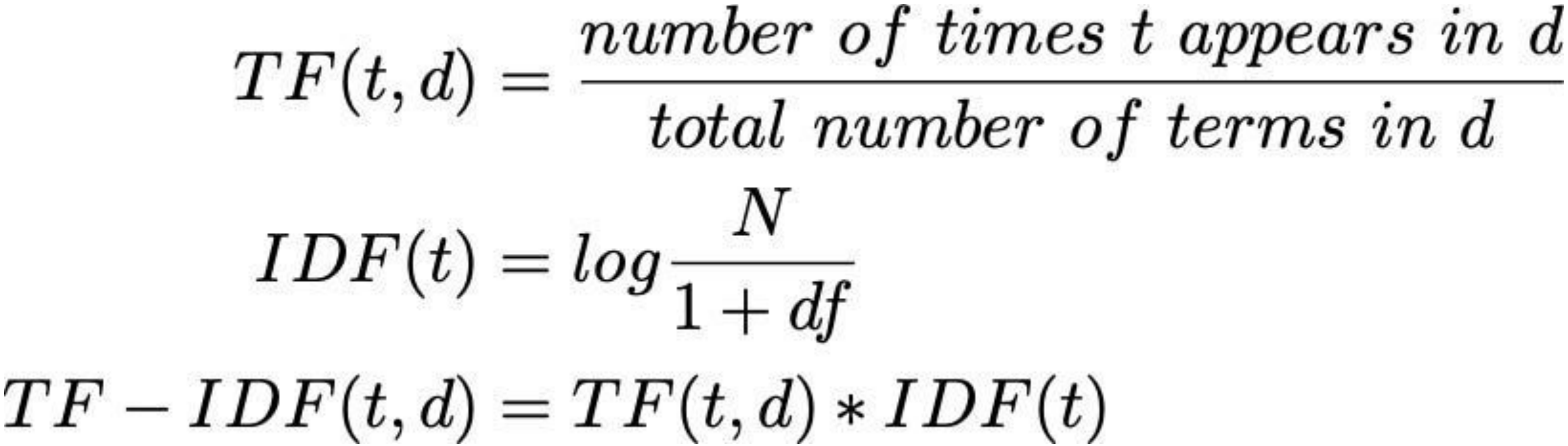
Becomes

Tokenize words and create vector for each survey record							
#	free	conference	best	ever	breakfast	only	came
1	1	1	0	0	0	0	0
2	0	1	1	1	0	0	0
3	0	0	1	1	1	0	0
4	1	1	0	0	1	1	1

What about context, order, punctuation, and frequency?

Term Freq., Inv. Document Frequency (TF-IDF)

Gives the relative importance of a term in a corpus (list of documents), given by the following formula below



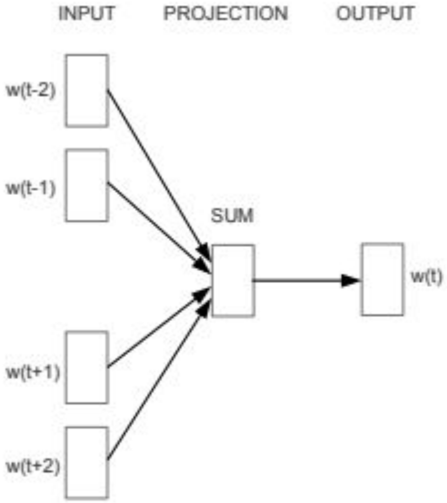
$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

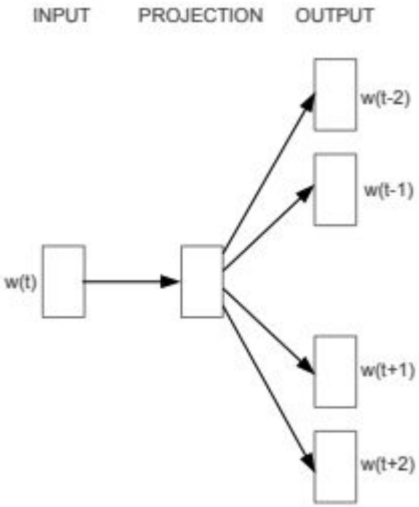
$$TF-IDF(t,d) = TF(t,d) * IDF(t)$$

Source:
https://kinder-chen.medium.com/introduction-to-natural-language-processing-tf-idf-1507e907c19

Word2Vec



Continuous Bag-of-Words “CBOW”
From context, predict word



Skip-gram
From word, predict context

Appendix / Project Discussion

Where to find dataset to work with?

1. https://paperswithcode.com/datasets
2. https://www.kaggle.com/datasets
3. https://datasetsearch.research.google.com/
4. https://archive.ics.uci.edu/ml/datasets.php
5. https://www.tensorflow.org/datasets
6. https://huggingface.co/docs/datasets/
7. https://registry.opendata.aws/
8. Everywhere …