A/B testing

Use statistic tools to compare two version of a web page for example headlines, fonts or any web products to see which one performs better. [Process](https://vwo.com/ab-testing/):

* study your web data
* observe customer behavior
* construct a hypothesis
* test your hypothesis
* analyze the data and draw conclusions
* report result

accuracy -> compare true positive and true negative.

[ACC](https://en.wikipedia.org/wiki/Confusion_matrix) = (TP+TN)/(P+N)

activation function

in deep learning, the activation function of a node defines the output of that node given an input or set if inputs.  
  
AdaGrad

AUC (Area under the ROC Curve)

ROC curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True positive rate, False positive rate.

[AUC](https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc) is an area, AUC provides an aggregate measure of performance across all possible classification thresholds.  
  
automation bias

Errors of automation bias tend to occur when decision-making is dependent on computers or otherautomated aids and the human is in an observatory role.

back-propagation

BP is a method used in artificial neural network to calculate a gradient that is needed in the weights to be used in the network  
  
baseline

A baseline result is the simplest possible prediction. It is a method that uses heuristics, simple summary statistics, randomness, of learning to create predictions for a dataset. For some problems, this maybe a randam result, and in others it may be the most common prediction.

batch

[Batch](https://www.quora.com/What-is-meant-by-Batch-in-machine-learning) means a group of traning samples, means how training samples are used while computing the loss function. It is distinct form “online” and “mini-batch” learning

batch size

number of training examples utilized/propagated in one iteration.

bias (math)

in probability, “bias” measures the possible outcomes are not equal to real value.

bias (ethics/fairness)

Bias is disproportionate weight in favor of or against one thing, person, or group compared with another. Usually in a way considered to be unfair.

binary classification

classify/group the elements of a given set into to groups/classes.

binning

[Binning](https://docs.tibco.com/pub/spotfire/7.0.1/doc/html/bin/bin_what_is_binning.htm) is a way to group a number of more or less continuous values into a smaller number of “bins”

bucketing

same as bining

calibration layer

candidate sampling

categorical data

Categorical data represents types of data which may be divided into groups. Basically it means the group belonging to.

centroid

the center of mass of the object, center of a cluster

checkpoint

class

Group  
class-imbalanced data set

The number of observations belonging to one class is significantly different than those belonging to the other class.

classification model

Build a [model](https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf) to assign objects to one of several predefined categories.   
  
classification threshold

Decision threshold, in order to map a logistic regression value to a binary category, toy must have such a classification threshold. A value above that threshold indicates a class, otherwise, it belongs to another class.

clustering

we don't know the classes before we build the model, we need to group similar objects into the same group.

collaborative filtering

[Collaborative filtering](https://towardsdatascience.com/various-implementations-of-collaborative-filtering-100385c6dfe0) is based on assumption that people like things similar to other things they like and things that are liked by other ppl with similar taste.  
  
confirmation bias

It is the tendency to search for, interpret, favor and recall information in a way that confirm one’s preexisting beliefs or hypo.

confusion matrix

It's a table used to measure the performance of model, we have real true and false value and predicted true and false value here.

continuous feature

a variable that has a infinite umber of possible values. For example time, distance, weight  
  
convergence

a state that reached during iteration in which training loss and validation loss change very little or even not at all with each iteration after a certain number of iterations.   
  
convex function

convex optimization

convex set  
  
[convolution](http://forums.fast.ai/t/dense-vs-convolutional-vs-fully-connected-layers/191)

applying a convolution filter to the input matrix which is usually has a much higher dimensionality than that of the convolution filter. Basically it turns a large matrix to a smaller one.

convolutional filter

a square matrix whose values have to be learned by a neural network training algorithm. Convolutional filters are parts of convolutional neural networks where they are used by multiplying their values by the values of the corresponding patches of the input (or the previous layer) and then summing the obtained values.

convolutional layer

convolutional neural network

A CNN is a class of feedforward neural network, most commonly applied to analyze images. It uses convolutional layer that filter inputs for useful information. convolutional layer have parameters that are learned so that filters are adjusted automatically to extract the most useful information for the task at hand.

Pooling also reduces the memory consumption and thus allows for the usage of more convolutional layers.

convolutional operation

Long story short, use a convolutional layer to scan a image to extract useful information.

cost

difference between target value and predictive value  
  
coverage bias

[Coverage bias](https://en.wikipedia.org/wiki/Survey_sampling) can occur when population members do not appear in the sample frame

cross-entropy

[Cross-entropy](https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html) loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0.

custom Estimator #TensorFlow

data analysis  
  
DataFrame

It is a 2-dimensional labeled data structure with columns of potentially different type.

dataset  
  
decision boundary

In a *classification* problem with two or more classes, a *decision boundary* is a hypersurface that partitions the underlying vector space into two or more regions, one for each *class*.

dense layer

[Dense layer](http://forums.fast.ai/t/dense-vs-convolutional-vs-fully-connected-layers/191/3) is a linear operation in which every input is connected to every output by a weight (so there are n\_inputs \* n\_outputs weights - which can be a lot!). Generally followed by a non-linear activation function

deep model

dense feature  
  
device #TensorFlow  
  
discrete feature  
  
dropout regularization

It prevents *neurons* from co-adapting by randomly setting a fraction of them to 0 at each training iteration.

dynamic model  
  
early stopping

*Early stopping* is a *regularization* method that involves ending model training before *training loss* finishes decreasing. In early stopping, the engineer ends model training when the *validation loss* starts to increase, that is when generalization performance worsens.

embeddings

An embedding maps an input representation, such as word, sentence or even images, into a vector. Most frequently embeddings refer to word embedding such as **word2vec** or **Glove**

empirical risk minimization (ERM)  
  
ensemble

learning a strong classifier by combining multiple weak classifier  
  
epoch

One passthrough the training set by a machine learning algorithm  
  
Estimator #tensorflow  
  
example #features

Also called instance is a member of a dataset, typically an example is a vector of features, each features represents some specific property of the example.   
  
experimenter's bias #fairness  
  
false negative (FN)

Model mistakenly predict the negative class, for example, the model inferred that a particular email message was not spam, but the email message actually was spam.

false positive (FP)

See above. mistakenly predict the positive clas.

feature

an attribute of an example, usually a part of a feature vector. if an example is a people, it can have the following features: height, weight, race, etc.

feature cross  
  
feature engineering  
  
feature set

feature selection

A progress of removing the dataset features that seem irrelevant for modeling.

feature spec #TensorFlow

feedforward neural network

a neural network wherein connections between the units do not form a cycle. As such, it is different from recurrent neural networks. The information in this network moves in only one direction, forward from the input nodes, through the hidden nodes and to the output nodes.  
  
few-shot learning

usually employed in *classification*, designed to learn effective classifiers from only a *small numbe*r of training examples.  
  
full softmax -> compare with candidate sampling.  
  
fully connected layer

gated recurrent unit (GRU)

a simplified version of an LSTM unit with **fewer parameters**. Just like an LSTM cell, it uses a gating mechanism to allow RNNs to efficiently learn long-range dependencies by reducing the vanishing gradient problem. The GRU consist of a **reset and update gate** that determine which the extent to which the unit should keep the old value and to which the new value has to replace it at the current time step.

generalization  
  
generalized linear model  
  
gradient  
  
gradient clipping

**capping gradient value** before applying them in gradient descent algorithm during backpropagation. Gradient clipping helps ensure numerical stability an **prevents exploding gradient problem**.  
  
gradient descent

**an iterative optimization algorithm for differentiable functions**. It is designed to find the **minimum** of a function. In many machine learning algorithms, gradient descent, or its variant is used to find the **minimum of the** **loss function** given the training dataset.

graph #TensorFlow

grid search

*Grid search* is a way of **hyperparameter tuning**. The process consists of training the *model* on all possible combinations of hyperparameter values and then selecting the best combination. The best combination of hyperparameters is the one that performs the best on the *validation set*.

group attribution bias -> comapre homogeneity bias and in-group bias.  
  
heuristic  
  
hidden layer

layers between input layers and output layers, where neurons take in a set of weighted inputs and produce an output through **an activation function**.

Hierarchical Clustering Algorithm

[*Hierarchical clustering algorithms*](https://semanti.ca/blog/?glossary-of-machine-learning-terms)is a category of *clustering algorithms* that create a tree of clusters. Hierarchical clustering algorithms are well-suited to hierarchical data, such as botanical taxonomies.

hinge loss

a loss function used for training classifiers. The hinge loss is used for “maximum-margin” classification, most notably for support vector machine.

Loss(y) = max(0, 1-y\*y^), y is the “raw” output of the classifier’s decision function, not the predicted class label.

holdout data

A part of the dataset that contains examples intentionally **not used during training**. **Validation set and test** set are examples of holdout data

hyperparameter

A parameter of a machine learning algorithm **whose value is not optimized during training**. It can be training iterations, the size of minibatch, a regularization parameter, the value of the learning rate, and many others.

Hyperparameter **can be optimized** using cross-validation and techniques like grid search, random search, Bayesian optimization, evolutionary optimization, and others.   
  
hyperplane

A boundary that **separates a space into two subspaces**. For example, a line is a hyperplane in two dimensions and a plane is a hyperplane in three dimensions. In machine learning, a hyperplane is usually a boundary separating a high-dimensional space.

implicit bias #fairness  
  
independently and identically distributed (i.i.d)  
  
inference  
  
in-group bias  
  
input function #TensorFlow  
  
input layer

a layer whose neurons take as **input the features of the input instance**.

instance

See examples  
  
interpretability  
  
inter-rater agreement  
  
iteration

**Number of times** the machine learning algorithm’s parameters are updated while training a model on a dataset.

k-means

clustering data into exactly k clusters.

First, define k initial cluster *centroids*.

Second, assign each example to the closest centroid.

Third, recompute the new position for each centroid as the average of the examples assigned to it.

Iterate back to step two.

K Nearest Neighbors

An *instance-based learning algorithm* that can be used for both classification and regression

When used in the *classification* context, the algorithm predicts the class of an *unlabeled example* as the majority class among kk its closest neighbors in the vector space. In the *regression* context, the label of an unlabeled example is calculated as an average of the labels of the kk its closest neighbors. The distance from one example to another is usually given by a *similarity metric*.

k-median  
  
Keras  
  
Kernel Support Vector Machines (KSVMs)

the function of kernel is to take data as input and transform it into the required form.

We need to create hyperspace to separate a space into subspace

For example,*linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.*

L1 loss

L1 regularization  
  
L2 loss  
  
L2 regularization  
  
label  
  
labeled example  
  
lambda

Regularization term

layer  
  
Layers API #TensorFlow  
  
learning rate

A **scalar** used to update model parameters via gradient descent.

Gradient step = gradient descent algorithm \* gradient by the learning rate

least squares regression  
  
linear regression  
  
logistic regression  
  
logits  
  
Log Loss

loss function used in the binary logistic regression

logloss(y) = -(ylog(p)+(1-y)log(1-p))

p is the probability predicted by the model, y is true label

Long short-term memory unit (LSTM)

Long short-term memory (LSTM) units in recurrent neural networks help reducing the vanishing gradient problem during the backpropagation. LSTM unit is a neuron that has a memory cell and three gates: "input", "output" and "forget". The purpose of the memory cell is to retain information previously used by the RNN or forget if needed. Neural networks with LSTM units, also called LSTM networks, are explicitly designed to avoid the long-term dependency problem in RNNs and have been shown to be able to learn complex sequences better than simple RNNs.

The structure of a memory cell is: an input gate, that determines how much of information from the previous layer gets stored in the cell; the output gate, that determines how of the next layer gets to know about the state of the current cell; and the forget gate, which determines what to forget about the current state of the memory cell.

log-odds  
  
loss function

In the *classification* context: describes the cost of assigning the *label* y^ to a sample of *class* y.

In the *regression* context: describes the cost of assigning the value y^ to the regression function evaluated at →xx→, where it takes value y.

machine learning

algorithms that learn from, make decisions and predictions based on data.

Mean Squared Error (MSE)  
  
metric

[Metrics](https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234) is for measuring the performance of our model.

Classification Accuracy

Logarithmic Loss

Confusion Matrix

Area under Curve

F1 Score

Mean Absolute Error

Mean Squared Error

mini-batch

see batch

mini-batch stochastic gradient descent (SGD)  
  
ML  
  
model

Statistical model is the result of a machine learning algorithm applied to the training data.

Model is often a parametrized mathematical formula, where parameters are learning by the machine learning algorithm.

model function #TensorFlow  
  
model training  
  
multi-class classification

*Multi-class classification* is a *classification* problem that distinguishes among more than two classes.

multinomial classification

Multinomial regression*is a variant of the*logistic regression*algorithm used for*multi-class classification*problems.*

negative class  
  
neural network  
  
neuron

also called unit is a node in a neural network, typically taking in multiple input values and generate one output value.

The neuron calculates the output value by applying an *activation function* (nonlinear transformation) to a weighted sum of input values.

node (neural network)  
  
node (TensorFlow graph)  
  
non-response bias  
  
normalization

the process of converting an actual range of values into a standard range of values, e.g [-1, 1], [0,1]

numerical data  
  
numpy  
  
objective  
  
one-hot encoding

transforming a **categorical feature** into a vector of several binary features where all components are 00, except for one component with a value of 1.

 if our *example* has a categorical feature "weather" and this feature has three possible values: "sun", "rain", "clouds", then to transform this feature into something a *machine learning algorithm* that can only work with numerical values, one can transform this feature into a vector of three numerical values:

sun=[1,0,0]

rain=[0,1,0]

clouds=[0,0,1]

one-shot learning

training a *model* with only a single *example* per *class*.

*One way to build a system capable of one-shot learning is to use*representation learning*, to learn representations or*features*of data that can be used to accurately classify a single example.*

one-vs.-all  
  
Operation (op) #TensorFlow  
  
optimizer  
  
momentum (Momentum)  
  
out-group homogeneity bias -> compare in-group bias.  
  
  
outliers

an example that appears far away and diverges from an overall pattern in the dataset.

output layer  
  
overfitting

perform very well in training data, but poor in testing data

*Overfitting* occurs when the machine learning algorithm learns a model that fits the training data too well by incorporating details and noise specific to the training data.

The problem of overfitting is usually solved by ***regularization*** or ***early stopping***,

pandas  
  
parameter

A *parameter* of a *model* is the quantity a machine learning algorithm modifies in order to minimize the *loss function*. For example, in *neural networks*, parameters are weights applied to inputs of *neurons*.

partial derivative  
  
participation bias  
  
performance  
  
perplexity  
  
pipeline  
  
pooling

reducing **a matrix created by an earlier convolution** to a smaller matrix. Pooling usually involves taking either the maximum or average value across the pooled area.

positive class -> Contrast with negative class.  
  
precision  
  
prediction

TP/(TP+FP), how many of these positive predictions were correct.

prediction bias  
  
premade Estimator #TensorFlow  
  
pre-trained model  
  
prior belief  
  
proxy labels

principle component analysis (PCA)

[PCA](https://semanti.ca/blog/?glossary-of-machine-learning-terms) is a linear transformation which projects n examples each consisting of m features on a hyperplane in such a way the projection error is minimal.

Basically, it is trying to find the most important part of a dataset to replace the whole giant dataset.

By keeping only several biggest principal components and the projections of the examples on them, and by discarding the rest of information, one can reconstruct the dataset from a much smaller amount of information (with some small loss in accuracy of reconstruction because of the discarded information).

queue #TensorFlow  
  
rank (Tensor)  
  
rank (ordinality)  
  
rater  
  
recall

TP/(TP+FN), measure how many of the positively labeled examples were correctly classified by the model.

Recurrent neural network RNN

A neural network that usually deal with sequential data like texts, audio and video.

It contains an internal memory. An RNN has an internal loop that allows information to persist in the network. *Neurons* receive information not just from the previous layer, but also from themselves from the previous pass. This means that the order of inputs to the RNN matter as RNNs have a state.

RNNs are particularly sensitive to the *vanishing* and *exploding gradient problems*, where depending on the *activation functions* used, the information can get lost over time. *Long short-term memory units* (LSTM) addresses this problem. RNNs are commonly used with sequential data, like in natural language processing.

Common examples of **sequential data** includes texts, audio and video.

regression model

a model that outputs continuous data.

regularization  
  
regularization rate

a technique to make the fitted function smoother. This helps to prevent overfitting. The most widely used regularization techniques are **L1**, **L2**, **dropout**, and **weight decay**.

reporting bias  
  
representation - features.  
  
root directory  
  
rotational invariance  
  
sampling bias  
  
scaling  
  
selection bias  
  
semi-supervised learning

learning a *model* by using both *labeled* and *unlabeled examples*.

Semi-supervised learning techniques take advantage of a small amount of labeled data and a large amount of unlabeled data to produce a better model than a purely supervised learning or a purely unsupervised learning technique.  
  
sequence model  
  
sigmoid function  
  
size invariance  
  
softmax  
  
sparse feature  
  
sparse representation  
  
sparsity  
  
spatial pooling  
  
squared hinge loss  
  
squared loss  
  
static model  
  
stationarity  
  
step  
  
step size  
  
stochastic gradient descent (SGD)

a type of *gradient descent* algorithm where the gradient of the function to be optimized is computed by taking a sample of data. The update to the coefficients is performed for each training instance, rather than at the end of the batch of instances.

The learning can be much faster with stochastic gradient descent for very large training datasets and often one only need a small number of passes through the dataset (one pass through the dataset is called *epoch*) to reach a good or good enough set of coefficients.

structural risk minimization (SRM)

Support Vector Machine (SVM)

 a classification algorithm that seeks to maximize the margin between positive and negative classes. SVM is often used together with kernels, functions that map input examples (given as multidimensional vectors) to a higher dimensional space.

Use **hinge loss** as loss function

stride  
  
subsampling  
  
summary in TensorFlow  
  
supervised machine learning  
  
synthetic feature  
  
target  
  
temporal data  
  
Tensor  
  
Tensor Processing Unit (TPU)  
  
Tensor rank  
  
Tensor shape  
  
Tensor size  
  
TensorBoard  
  
TensorFlow  
  
TensorFlow Playground  
  
TensorFlow Serving  
  
test set  
  
tf.Example  
  
time series analysis

Tokenization

the process of splitting a text string into units called tokens. A token may be a word or a group of words.

training  
  
training set  
  
transfer learning

Using a model trained to solve one problem to help to solve another problem.

For example, a *neural network* trained to distinguish between different kinds of jungle animals can be reused to train another neural network that would distinguish between different kinds of domestic animals.

translational invariance  
  
true negative (TN)  
  
true positive (TP)  
  
true positive rate (TP rate)

Underfitting

a situation in which the *model* trained on the *training data* **doesn't predict well *training examples***.

unlabeled example  
  
unsupervised machine learning -> compare with supervised machine learning.

*Unsupervised learning* is a problem, given an *unlabeled dataset*, automatically find **hidden (or latent) structure in this dataset**.

validation loss

*Validation loss* is the average loss computed by applying the *loss function* the outputs of the model applied to the examples from the validation set.

validation set

The *validation set* is a holdout set used for *hyperparameter tuning*.

Vanishing Gradient Problem

The *vanishing gradient problem* happens in very deep neural networks, typically recurrent neural networks, that use activation functions whose gradients tend to be small. Because these small gradients are multiplied during backpropagation, they tend to "vanish" throughout the layers, preventing the network from learning long-term dependencies. Common ways to counter this problem is to use activation functions like *ReLU* or *LSTM* that do not suffer from small gradients. The opposite of this problem is called the *exploding gradient problem*.

Variance

Model contains so many noises of training model.

an error from sensitivity to small fluctuations in the *training set*. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (overfitting).

weight  
  
wide model

zero shot learning

Zero shot learning is the problem of learning a model capable of classifying examples whose labels weren't present in the training data. Usually, a solution involves embedding both *feature vectors* and class labels. For example, if the problem is to classify animals by their pictures, the training set can not contain an image of a dog. However, the classifier will predict the class embedding which can be used to find the corresponding label by looking in the embedding-label lookup table. The most frequently used embeddings for class labels are word2vec. However, one can train its own embedding for class labels in the case when the label can be multi-word.

plus: (term explanation, example/deployment, use experience)

* background
* detail of projects
* **data sources**
* version of spark
* plotting library in Scala
* RNN, LSTM, CNN, FNN
* Deep learning models utilized
* Bidirectional LSTM and LSTM
* LSTM & GRU
* PCA & autoencoder
* Prevent overfitting with neural network
* Machine learning pipeline
* Serverless architecture VS traditional systems
* **Deployment of ML in a production environment (to prove you actually worked before) Architecture**
* **Docker** experience
* **Deployment experience in Spark, sparkml, v.s tensorflow? v.s keras?**
* Apache Kafka / producer / intermediate stage
* Standardization of a schema
* Scale up? handle larger size of data
* Collaborate with ppl
* Pre-processing data
* Data source
* QA environment
* Best package for a Hadoop (r?)
* R shiny
* R-packages: Gam and mgcv
* Union and union all
* Truncate and delete
* Show hidden directories / ls commands `ls -ad .\*`
* RJDBC
* Apache Drill
* Structured, semi structured and unstructured data
* Zaloni
* NLP project?
* Bias and variance
* POS tagging
* NER pipeline
* productionize models
* Jenkins
* Engineering side
* Continuous integration

Update model to fit new data

* Leadership (hands on / off? 自己干活么)