1 、 机器学习好文章

（1）13种主流机器学习的框架：

<https://blog.csdn.net/zuochao_2013/article/details/77852442>

（2）[机器学习通用框架](https://www.cnblogs.com/harvey888/p/5966637.html)

<https://www.cnblogs.com/harvey888/p/5966637.html>

（3）主流深度学习框架和通用机器学习框架对比

<https://blog.csdn.net/baihuaxiu123/article/details/54580117/>

（4）最全Tensorflow2.0 入门教程持续更新

<https://zhuanlan.zhihu.com/p/59507137>

2 、 机器学习中常见的几种损失函数

参考：<https://www.cnblogs.com/hejunlin1992/p/8158933.html>

1. 0-1损失函数和绝对值损失函数；

0-1损失是指，预测值和目标值不相等为1，否则为0：



绝对值损失函数：



1. log对数损失函数



1. 平方损失函数



1. 指数损失函数



1. Hinge损失函数

Hinge loss用于最大间隔（maximum-margin）分类，其中最有代表性的就是支持向量机SVM



3 、 常用的模型评价指标有哪些？

参考：<https://www.cnblogs.com/inchbyinch/p/12622667.html>

混淆矩阵、准确率、（宏/微）查准率、查全率、F1指数、PR曲线、ROC曲线/AUC；

4 、 常用python机器学习相关的库有哪些

参考：<https://zhuanlan.zhihu.com/p/44952874>

<https://blog.csdn.net/weixin_34004576/article/details/93014855>

Tensorflow、Keras、Theano、PyTorch、LightGBM、Numpy、Pandas、SciPy、Scikits\_Learn、Eli5；

5 、 特征向量的缺失值处理有哪些方法？

参考：https://blog.csdn.net/w352986331qq/article/details/78639233

1. 删除

主要有简单删除法和权重法。简单删除法是对缺失值进行处理的最原始方法。

(1) 简单删除法

此方法将存在缺失值的数据条目（对象，元组，记录）进行删除。

(2) 权重法

当缺失值的类型为非完全随机缺失的时候，可以通过对完整的数据加权来减小偏差。

2. 填补

这类方法是用一定的值去填充空值，从而使信息表完备化。通常基于统计学原理，根据决策表中其余对象取值的分布情况来对一个空值进行填充，譬如用其余属性的平均值来进行补充等。数据挖掘中常用的有以下几种补齐方法：

(1) 人工填写（filling manually）

由于最了解数据的还是用户自己，因此这个方法产生数据偏离最小，可能是填充效果最好的一种。然而一般来说，该方法很费时，当数据规模很大、空值很多的时候，该方法是不可行的。

(2) 特殊值填充（Treating Missing Attribute values as Special values）

将空值作为一种特殊的属性值来处理，它不同于其他的任何属性值。如所有的空值都用“unknown”填充。这样将形成另一个有趣的概念，可能导致严重的数据偏离，一般不推荐使用。

(3) 均值填充（Mean/Mode Completer）

将信息表中的属性分为数值属性和非数值属性来分别进行处理。如果空值是数值型的，就根据该属性在其他所有对象的取值的平均值来填充该缺失的属性值；如果空值是非数值型的，就根据统计学中的众数原理，用该属性在其他所有对象的取值次数最多的值(即出现频率最高的值)来补齐该缺失的属性值。另外有一种与其相似的方法叫条件平均值填充法（Conditional Mean Completer）。在该方法中，缺失属性值的补齐同样是靠该属性在其他对象中的取值求平均得到，但不同的是用于求平均的值并不是从信息表所有对象中取，而是从与该对象具有相同决策属性值的对象中取得。

(4) 热卡填充（Hot deck imputation，或就近补齐）

对于一个包含空值的对象，热卡填充法在完整数据中找到一个与它最相似的对象，然后用这个相似对象的值来进行填充。不同的问题可能会选用不同的标准来对相似进行判定。该方法概念上很简单，且利用了数据间的关系来进行空值估计。这个方法的缺点在于难以定义相似标准，主观因素较多。

(5) 聚类填充(clustering imputation)

最为典型的代表是K最近距离邻法（K-means clustering），先根据欧式距离或相关分析来确定距离具有缺失数据样本最近的K个样本，将这K个值加权平均来估计该样本的缺失数据。同均值插补的方法都属于单值插补，不同的是，它用层次聚类模型预测缺失变量的类型，再以该类型的均值插补。

(6) 使用所有可能的值填充（Assigning All Possible values of the Attribute）

这种方法是用空缺属性值的所有可能的属性取值来填充，能够得到较好的补齐效果。

(7) 组合完整化方法（Combinatorial Completer）

这种方法是用空缺属性值的所有可能的属性取值来试，并从最终属性的约简结果中选择最好的一个作为填补的属性值。

(8) 回归（Regression）

基于完整的数据集，建立回归方程（模型）。对于包含空值的对象，将已知属性值代入方程来估计未知属性值，以此估计值来进行填充。

(9) 极大似然估计（Max Likelihood ，ML）

在缺失类型为随机缺失的条件下，假设模型对于完整的样本是正确的，那么通过观测数据的边际分布可以对未知参数进行极大似然估计（Little and Rubin）。

(10) 多重插补（Multiple Imputation，MI）

多值插补的思想来源于贝叶斯估计，认为待插补的值是随机的，它的值来自于已观测到的值。具体实践上通常是估计出待插补的值，然后再加上不同的噪声，形成多组可选插补值。根据某种选择依据，选取最合适的插补值。

3. 不处理

直接在包含空值的数据上进行数据挖掘。这类方法包括贝叶斯网络和人工神经网络等。

6 、 归一化、标准化、正则化的区别

参考：https://blog.csdn.net/tianguiyuyu/article/details/80694669

1、归一化（Normalization）：

（1）把数据变为（0，1）之间的小数。主要是为了方便数据处理，因为将数据映射到0～1范围之内，可以使处理过程更加便捷、快速。

（2）把有量纲表达式变换为无量纲表达式，成为纯量。经过归一化处理的数据，处于同一数量级，可以消除指标之间的量纲和量纲单位的影响，提高不同数据指标之间的可比性。

2、标准化（Standardization）：

数据的标准化是将数据按比例缩放，使之落入一个小的特定区间。

3、正则化（Regularization）：

用一组与原不适定问题相“邻近”的适定问题的解，去逼近原问题的解，这种方法称为正则化方法。如何建立有效的正则化方法是反问题领域中不适定问题研究的重要内容。通常的正则化方法有基于变分原理的Tikhonov 正则化、各种迭代方法以及其它的一些改进方法。

总的来说，归一化是为了消除不同数据之间的量纲，方便数据比较和共同处理，比如在神经网络中，归一化可以加快训练网络的收敛性；标准化是为了方便数据的下一步处理，而进行的数据缩放等变换，并不是为了方便与其他数据一同处理或比较，比如数据经过零-均值标准化后，更利于使用标准正态分布的性质，进行处理；正则化而是利用先验知识，在处理过程中引入正则化因子(regulator)，增加引导约束的作用，比如在逻辑回归中使用正则化，可有效降低过拟合的现象。

7 、 特征向量的归一化方法有哪些？

参考：https://blog.csdn.net/zhangmengjlu/article/details/9844949

1、线性函数转换，表达式如下：

y=(x-MinValue)/(MaxValue-MinValue)

2、对数函数转换，表达式如下：

y=log10 (x)

3、反余切函数转换 ，表达式如下：

y=arctan(x)\*2/PI

4、减去均值，乘以方差：

y=(x-means)/ variance

8 、 哪些机器学习算法不需要做归一化处理？

参考：https://zhuanlan.zhihu.com/p/57130209

在实际应用中，通过梯度下降法求解的模型一般都是需要归一化的，比如线性回归、logistic回归、KNN、SVM、神经网络等模型。

但树形模型不需要归一化，因为它们不关心变量的值，而是关心变量的分布和变量之间的条件概率，如决策树、随机森林(Random Forest)。

9 、 为什么一些机器学习模型需要对数据进行归一化？

参考：https://blog.csdn.net/xbmatrix/article/details/56695825

1）归一化后加快了梯度下降求最优解的速度；

机器学习模型使用梯度下降法求最优解时，归一化往往非常有必要，否则很难收敛甚至不能收敛。

2）归一化有可能提高精度：

一些分类器需要计算样本之间的距离（如欧氏距离），例如KNN。如果一个特征值域范围非常大，那么距离计算就主要取决于这个特征，从而与实际情况相悖（比如这时实际情况是值域范围小的特征更重要）。

10 、 特征比数据量还大时，选择什么样的分类器？

参考：https://www.zhihu.com/question/32284740

特征比数据还多，意味着整体来看，数据是稀疏的，

把特征视为维度，高维度下稀疏的数据很有可能是线性可分的，

综上，选择线性分类器

11 、 简单介绍一下ID3、C4.5、CART原理，区别与联系

参考：https://blog.csdn.net/lzzdflg/article/details/78649925

联系：

CART与ID3和C4.5相同都由特征选择，树的生成，剪枝组成。但ID3和C4.5用于分类，CART可用于分类与回归。

区别：

CART是在给定输入随机变量X条件下输出随机变量Y的条件概率分布，与ID3和C4.5的决策树所不同的是，ID3和C4.5生成的决策树可以是多叉的，每个节点下的叉树由该节点特征的取值种类而定，比如特征年龄分为（青年，中年，老年），那么改节点下可分为3叉。而CART为假设决策树为二叉树，内部结点特征取值为”是”和”否”。左分支取值为”是”，有分支取值为”否”。这样的决策树等价于递归地二分每一个特征，将输入空间划分为有限个单元，并在这些单元上预测概率分布，也就是在输入给定的条件下输出条件概率分布。

12 、 简单介绍一下决策树  
参考：https://www.cnblogs.com/liwenqiao/p/5424517.html

决策树学习是一种采用树状结构的有监督机器学习方法。决策树是一个预测模型，表示对象特征和对象值之间的一种映射。

简而言之，利用决策树进行预测的过程就是从根节点开始，根据样本的特征属性选择不同的分支，直到到达叶子结点，得出预测结果的过程。

决策树优点：

1）.决策树模型可以读性好，具有描述性，有助于人工分析；

2）.效率高，决策树只需要一次构建，反复使用，每一次预测的最大计算次数不超过决策树的深度（特征均为离散特征）。

13 、 决策树的优点及缺点有哪些？

参考：https://www.dongao.com/zckjs/gsz/201909031142848.shtml

决策树法优点：

决策树列出了决策问题的全部可行方案和可能出现的各种自然状态，以及各可行方法在各种不同状态下的期望值。能直观地显示整个决策问题在时间和决策顺序上不同阶段的决策过程。在应用于复杂的多阶段决策时，阶段明显，层次清楚，便于决策机构集体研究，可以周密地思考各种因素，有利于作出正确的决策。

决策树法缺点：

使用范围有限，无法适用于一些不能用数量表示的决策；对各种方案的出现概率的确定有时主观性较大，可能导致决策失误

14 、 简单介绍一下CART决策树

参考：https://www.cnblogs.com/nxld/p/6170931.html

https://blog.csdn.net/moxiaoxuan123/article/details/81414479

CART决策树又称分类回归树，当数据集的因变量为连续性数值时，该树算法就是一个回归树，可以用叶节点观察的均值作为预测值；当数据集的因变量为离散型数值时，该树算法就是一个分类树，可以很好的解决分类问题。但需要注意的是，该算法是一个二叉树，即每一个非叶节点只能引伸出两个分支，所以当某个非叶节点是多水平(2个以上)的离散变量时，该变量就有可能被多次使用。

15 、 随机森林有哪些优缺点？

参考：https://blog.csdn.net/xingxing1839381/article/details/81273793

http://www.cnitedu.cn/it/share/20197676.html

随机森林的优点：

* + 可以处理高维数据，不同进行特征选择（特征子集是随机选择）
  + 模型的泛化能力较强
  + 训练模型时速度快，成并行化方式，即树之间相互独立
  + 模型可以处理不平衡数据，平衡误差
  + 最终训练结果，可以对特种额排序，选择比较重要的特征
  + 随机森林有袋外数据（OOB），因此不需要单独划分交叉验证集
  + 对缺失值、异常值不敏感
  + 模型训练结果准确度高
  + 相对Bagging能够收敛于更小的泛化误差

随机森林的缺点：

* + 当数据噪声比较大时，会产生过拟合现象
  + 对有不同取值的属性的数据，取值划分较多的属性会对随机森林产生更大的影响

16 、 随机森林的“随机”体现在哪些方面？

参考：https://blog.csdn.net/beyond\_2016/article/details/80011403

随机森林的随机性体现在每颗树的训练样本是随机的，树中每个节点的分裂属性集合也是随机选择确定的。有了这2个随机的保证，随机森林就不会产生过拟合的现象了。

17 、 如何对决策树进行剪枝？

参考：https://www.cnblogs.com/yonghao/p/5064996.html

两种方案：先剪枝和后剪枝

先剪枝说白了就是提前结束决策树的增长，跟上述决策树停止生长的方法一样。

后剪枝是指在决策树生长完成之后再进行剪枝的过程。这里介绍三种后剪枝方案：

（1）REP—错误率降低剪枝

顾名思义，该剪枝方法是根据错误率进行剪枝，如果一棵子树修剪前后错误率没有下降，就可以认为该子树是可以修剪的。

REP剪枝需要用新的数据集，原因是如果用旧的数据集，不可能出现分裂后的错误率比分裂前错误率要高的情况。由于使用新的数据集没有参与决策树的构建，能够降低训练数据的影响，降低过拟合的程度，提高预测的准确率。

（2）PEP—悲观剪枝

悲观剪枝认为如果决策树的精度在剪枝前后没有影响的话，则进行剪枝。怎样才算是没有影响？如果剪枝后的误差小于剪枝前经度的上限，则说明剪枝后的效果与剪枝前的效果一致，此时要进行剪枝。

18 、 Xgboost与GBDT的区别与联系是什么？

参考：https://blog.csdn.net/qq\_29462849/article/details/89421933

* GBDT和XGBoost的不同点
* GBDT是机器学习算法，XGBoost是该算法的工程实现。
* 在使用CART作为基分类器的时候，XGBoost显式地加入了正则项来控制模型的复杂度，有利于防止过拟合，从而提高模型的泛化能力。
* GBDT在模型训练时只使用了代价函数的一阶导数信息，XGBoost对代价函数进行二阶泰勒展开，可以同时使用一阶和二阶导数。
* 传统的GBDT采用CART作为基分类器，XGBoost支持多种类型的基分类器，比如线性分类器。
* 传统的GBDT在每轮迭代时使用全部的数据，XGBoost则采用了与随机森林相似的策略，支持对数据进行采样。
* 传统的GBDT没有设计对缺失值进行处理，XGBoost能够自动学习出缺失值的处理策略。

19 、 介绍一下GBDT的基本原理

参考：https://blog.csdn.net/fenghuibian/article/details/90741739

GBDT是一个基于迭代累加的决策树算法，它通过构造一组弱的学习器（树），并把多棵决策树的结果累加起来作为最终的预测输出。

20 、 XGBoost有什么优点

参考：https://blog.csdn.net/qq\_35290785/article/details/97564676

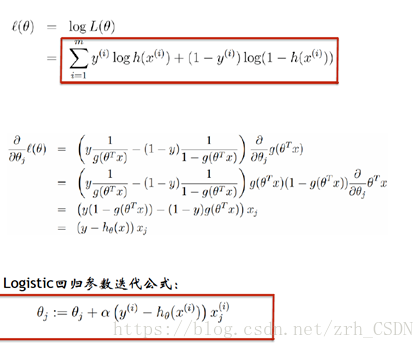
主要优点

* 正则化项​​​​​​​防止过拟合
* xgboost不仅使用到了一阶导数，还使用二阶导数，损失更精确，还可以自定义损失
* XGBoost的并行优化，XGBoost的并行是在特征粒度上的​​​​​​​
* 考虑了训练数据为稀疏值的情况，可以为缺失值或者指定的值指定分支的默认方向，这能大大提升算法的效率
* 支持列抽样，不仅能降低过拟合，还能减少计算​​​​​​​

21 、 常用的损失函数有哪些？(重复)

22 、 LR的原理和Loss的推导

参考：<https://blog.csdn.net/zrh_CSDN/article/details/80934278>



23 、 RF与GBDT之间的区别

参考：https://blog.csdn.net/qq\_19446965/article/details/81591543

相同点：都是由多棵树组成，最终的结果都是由多棵树一起决定。

不同点：

* 随机森林采用bagging，而GBDT采用boosting
* 组成随机森林的树可以分类树也可以是回归树，而GBDT只由回归树组成
* 组成随机森林的树可以并行生成，而GBDT是串行生成
* 随机森林的结果是多数表决表决的，而GBDT则是多棵树累加之和
* 随机森林对异常值不敏感，而GBDT对异常值比较敏感
* 随机森林是减少模型的方差，而GBDT是减少模型的偏差
* 随机森林不需要进行特征归一化。而GBDT则需要进行特征归一化
* 随机森林对训练集一视同仁，GBDT是基于权值的弱分类器的集成

24 、 L1和L2正则的区别，如何选择L1和L2正则

参考：https://zhuanlan.zhihu.com/p/30147914

L1是在loss function后面加上模型参数的1范数（也就是|xi|）L0范数的最小化问题在实际应用中是NP难问题，无法实际应用。L2是在loss function后面加上模型参数的2范数（也就是sigma(xi^2)），注意L2范数的定义是sqrt(sigma(xi^2))，在正则项上没有添加sqrt根号是为了更加容易优化。

L1 会产生稀疏的特征，L2 会产生更多地特征但是都会接近于0。L1在特征选择时候非常有用，而L2就只是一种规则化而已。具体的L1为什么会产生稀疏的特征，请看这里，说的很详细。

L1对应拉普拉斯分布，L2对应高斯分布。

L1不可导可以使用Proximal Algorithms或者ADMM来解决。

25 、 L1范式和L2范式的区别

26 、 什么是过拟合，产生的原因，以及解决的方法有哪些？

27 、 什么是过拟合？如何解决过拟合

28 、 解决过拟合的一种方法是增加数量集，如何增加数据集呢？

29 、 为什么要将求解SVM的原始问题转换为其对偶问题？

30 、 SVM为什么采用间隔最大化？

31 、 SVM的原理是什么？

32 、 SVM如何处理多分类问题？

33 、 SVM与树模型之间的区别

34 、 SVM的模型的推导

35 、 聚类相关面试题

36 、 机器学习常用的距离度量方法有哪些？

37 、 基于内容推荐算法有什么优缺点

38 、 常见的推荐算法有哪些

39 、 机器学习中有哪些优化方法

40 、 有监督学习和无监督学习有什么区别

41 、 收集的AI、机器学习面试题

42 、 机器学习常用的评价指标有哪些？

43 、 集成学习中的Bagging方法和Boosting方法有什么区别？

44 、 集成学习有哪些方法，各有什么特点？

45 、 对于维度极低的特征，你是选择线性还是非线性分类器？

46 、 线性分类器与非线性分类器的区别以及优劣

47 、 什么是生成模型和判别模型？

48 、 什么是"泛化能力"?

49 、 有监督学习和无监督学习的区别

50 、 机器学习项目的一般流程？

51 、 请描述梯度消失和梯度膨胀

52 、 列举优化算法及其优缺点