1. 异常值检测方法：
   - 数字异常值(Numeric outlier),使用IQR，适用于1维空间，一般k取1.5
   - Z-score，适用于1维或者低维，假定数据服从高斯分布，对数据做归一化，判断异常点方法一般设置为2.5/3/3.5
   - DBSCAN，基于DBSCAN聚类方法，适用于1维多维，基于数据密度的离群值的检验方法，主要将数据点分为核心点，边界点(在核心点ε内但不是核心点)，除此之外的都是噪声点
   - Isolation Forest，一维或多维特征空间中大数据集的非参数方法，基于孤立数进行判断数据点是否是异常点，孤立数越小越可能是异常点
2. 处理异常值的方法
   - 缺失值太多直接删除
   - 缺失值不多，可以考虑填充，例如均值填充(正态分布)，众数或中位数填充(长尾分布)
   - 将缺失值视为一个类别
   - 对于某些模型可以不处理，例如树模型(sklearn中只有xgboost自己有处理的方式而已其他没有)
3. 不同模型举例
   - 判别模型: KNN, LR, SVM, 树模型, 神经网络
   - 生成模型: Naive Bayes, GMM, HMM, LDA
4. 各个模型的三要素
   线性回归
   – 损失函数: 平方损失
   – 优化算法: 梯度下降
   LR:
   – 损失函数: 对数损失
   – 优化算法: 梯度下降
   SVM:
   – 损失函数: hinge loss
   – 优化算法: 凸优化，SMO
   Adaboost:
   – 损失函数: 指数损失
   – 优化算法: 对错分类样本加大权重
   决策树:
   – 损失函数: 节点个数 * 节点熵 + 树节点个数
   – 优化算法: 优化算法不准确，对于决策树说是构建方式，最优划分属性依据包括信息增益，信息增益率，gini index等。至于预减枝后减枝是泛化方法
   xgboost:
   – 损失函数: 多种，只要有二阶导都可以，视具体问题定
   – 优化算法: 牛顿法
   感知机:
   – 损失函数：各点的函数距离的负数(–Σyi*(w*x+b))
   – 优化算法: 随机梯度下降(仅仅是误分类点)
5. 评估方法
   - leave-one-out: 适用于数据量较大
   - K-Fold: 计算量比较大, 适用于数据量不大的时候
   - bootstrap: 小样本适用, 但会使得数据分布改变
     注意采样的时候需要根据问题进行判断是否需要进行分层采样
6. 评估准则
   回归
   – MSE
   – MAE
   – R2: 表征模型对于观察值的拟合程度
   – RMSE:
   – MAPE: MAPE=(Σ((X–Y)/X)*100%)/N，X为实测值，Y为模拟值，N为样本总数, 查看比例
   二分类
   – accuracy: 不适用于非均衡数据
   – ROC: 适用于非均衡数据, 关注样本rank值不在意绝对值大小
   – PR curve: 主要适用于对precision或者recall敏感的问题
   – Fß score: 带权重的PR的调和平均, ß>1 recall占权重大, 0<ß<1 precision占权重大
   多分类
   – micro AUC/F1
   – macro AUC/F1，主要区别在于平均的粒度层次不同
7. LR损失函数MSE能用吗？与极大似然之间关系？使用的优化算法是什么？learning rate的影响？
   - LR的损失函数是对数损失，不能使用MSE，因为如果使用MSE损失函数是非凸函数，难以优化。
   - LR是一个二分类模型，本质上是一个0–1分布，如果想求最优参数，可以通过写出似然函数取对数之后最大化似然函数即可，当对似然函数取负数，既可作为相应的LR的损失函数。
   - 使用的优化算法是梯度下降算法。
   - learning rate如果取得太大会导致损失函数震荡，难以收敛，learning rate如果太小会导致迭代次数增加收敛时间变长。
8. LR/SVM/softmax/Adaboost损失函数之间的差别?
   softmax模型是LR模型从二分类推广到多分类的模型，两者损失函数都是对数损失。SVM使用的是合页损失，分类边界只受限于支持向量。Adaboost损失函数是指数损失。
9. LR/DT/SVM模型之间对比
   LR是一个分类模型，DT和SVM既可以做分类也可以做回归。LR的决策面与线性SVM的决策面基本相同，原理上来说，分类问题SVM的决策面由支持向量决定，LR的决策面是由所有数据点决定，更进一步的说，LR不同数据点对决策面的贡献是不同的(由参数更新函数决定)，离决策面近的贡献大，因此SVM相对更加健壮，LR的决策面可能会受到正负样本点的分布的影响。DT与两者之间的差别在于决策面是阶梯型的，这是因为本质上说决策树进行决策时，样本空间是由各个特征的某个值进行划分的因此平行于特征轴，所以导致决策面是这种形态
10. SVM有哪些kernel？kernel的形态是怎样的？什么情况下选择什么样的kernel？满足什么样的条件可做为kernel function？RBF为什么可以映射到无限高维?
    - SVM主要有线性核，多项式核，高斯核，拉普拉斯核，sigmoid核
    - 形式查看寒老师SVM课件
    - 一般情况下如果数据量和特征都足够多时，非线性与线性核差不多；当数据量足够多，特征比较少的时候使用非线性核；当数据量比较少，特征比较多的时候使用线性核
    - 需要满足Mercer定理，即对于训练样例其核函数应该是对称半正定的。
    - 可以通过泰勒展开映射到无穷多维
11. SVM对缺失点和异常点是否敏感?
    对缺失值敏感，没有树模型那种处理缺失值的能力，对异常点不敏感，因为决策边界只取决于支持向量。
12. DT模型中ID3, C4.5, CART之间的差异?
    ID3使用的树的分支准则是信息增益，C4.5是在此基础上进行改经使用信息增益率，是为了防止选到某些熵比较大的属性(即属性取值过于分散)，CART使用的标准是基尼指数。ID3和C4.5算法分支的时候都是多叉树，CART是二叉树。CART除了分类还可以处理回归问题，对于连续值处理的时候，可以以某属性相邻值之间的均值作为分支条件。
13. 树模型对于缺失值如何处理?
    –RF
    首先，给缺失值预设一些估计值，比如数值型特征，选择其余数据的中位数或众数作为当前的估计值，然后，根据估计的数值，建立随机森林，把所有的数据放进随机森林里面跑一遍。记录每一组数据在决策树中一步一步分类的路径，然后来判断哪组数据和缺失数据路径最相似，引入一个相似度矩阵，来记录数据之间的相似度，比如有N组数据，相似度矩阵大小就是N*N，如果缺失值是类别变量，通过权重投票得到新估计值，如果是数值型变量，通过加权平均得到新的估计值，如此迭代，直到得到稳定的估计值。
    - xgboost
      xgboost把缺失值当做稀疏矩阵来对待，本身的在节点分裂时不考虑的缺失值的数值。缺失值数据会被分到左子树和右子树分别计层损失，选择较优的那一个。如果训练中没有数据缺失，预测时出现了数据缺失，那么默认被分类到右子树
14. RF与boosting之间差别？GBDT和xgboost之间差别?
    - RF是一种以决策树为基模型的bagging算法，RF通过对样本以及特征采样构建数据集，再利用DT进行集成，是一种天然的并行模型。
      boosting是一种串行集成方法，主要的思想是通过每一次增加一个基学习器来减少集成模型与预测值之间的误差，代表的模型包括Adaboost,GBDT以及xgboost，Adaboost是通过对样本点的权值调整(错误样本权值增大，正确样本权值降低)达到训练多个模型的目的。
    - GBDT与xgboost主要区别分为以下几点
      ①GBDT优化方法使用了一阶导数xgboost使用了二阶导数(牛顿法)优化
      ②xgboost模型本省增加了正则化项，包括叶子节点的二范数以及树结构的惩罚项
      ③xgboost本身有一些并行化的操作，对特征进行分bin求优再汇总求优
      ④xgboost使用了shrinkage方法，降低每次基学习器的拟合效果，防止过拟合
      ⑤xgboost支持特征抽样
      ⑥xgboost可对特征有缺失的样本自动处理
      ⑦可以对特征预排序进而使得特征选择的时候可以并行化价快速度
      ⑧除了CART还支持线性分类器
15. L1/L2正则化之间的差别？以及原因?
    - L1正则化具有截断效应，L2正则化具有缩放效应，换句话说L1正则化具有特征选择的作用。
    - 可以从两个方面来解释：
      ① L1和L2分别是求有约束条件下的函数损失函数的最优值，L1的约束条件是正方形的，当取得最优值时与坐标轴的相交的概率较大，L2约束条件是圆形的，损失函数取得最优值与坐标轴橡胶的概率较低。
      ② 从贝叶斯的角度来解释，L1正则相当于对参数加了一个拉普拉斯分布的先验，L2相当于对参数加了一个高斯分布的先验，拉普拉斯分布的特点就是在参数为0出的概率密度极大，而高斯分布的特点是参数在0附近的概率密度都很大，因此L1更容易使得参数变为0(即筛除特征)
16. SVM算法原始问题转化为对偶问题指的是什么？为什么需要将原始问题转化为对偶问题？什么场景下使用?
    - 原始问题指的是先对拉格朗日乘子求最大值在对w和b求最小值，对偶问题求解顺序是相反的
    - 原始问题的算法复杂度是与特征维度(w的维度)相关的，对偶问题的算法复杂度是与样本维度(拉格朗日乘子维度)相关的，其实可以不转化，之所以要转换是因为特征维度比样本维度要高。
    - 转化为对偶问题的使用场景一般都是非线性分类，如：使用高斯核函数将特征映射到无穷维。一般来说使用线性核进行分类解决原始问题即可。
17. Kmeans算法中k的确定？kmeans的优缺点?
    - 一般有以下几种方法
      ①手肘法，计算SSE(点在内部的平均距离)
      ②轮廓系数，(点到最近簇平均距离–点到自己簇平均距离)/max(点到最近簇平均距离,点到自己簇平均距离)
      ②Gap statistics，越大越好
    - 算法优点：
      ①算法简单
      ②只需要调k
    - 算法缺点
      ①结果受初值影响不稳定，解决方法多次初始取最优，kmeans++
      ②受异常点影响大，可以使用k–medias算法
      ③数据如果不是凸集效果较差，因为假设数据服从正态分布，是GMM模型的特殊形式。
18. xgboost与lightgbm之间差别?
    - XGBoost使用基于预排序的决策树算法，每遍历一个特征就需要计算一次特征的增益，时间复杂度为O(datafeature)。而LightGBM使用基于直方图的决策树算法，直方图的优化算法只需要计算K次，时间复杂度为O(Kfeature)
    - XGBoost使用按层生长(level–wise)的决策树生长策略，LightGBM则采用带有深度限制的按叶子节点(leaf–wise)算法。在分裂次数相同的情况下，leaf–wise可以降低更多的误差，得到更好的精度。leaf–wise的缺点在于会产生较深的决策树，产生过拟合。
    - 支持类别特征，不需要进行独热编码处理
    - 优化了特征并行和数据并行算法，除此之外还添加了投票并行方案（topk 再进行选择）
    - 采用基于梯度的单边采样来保持数据分布，减少模型因数据分布发生变化而造成的模型精度下降（梯度小的样本随机采样）
    - 特征捆绑转化为图着色问题，减少特征数量?
19. 什么是overfiting underfiting？如何判断？怎么缓解?
    - overfiting指的是在训练集效果极好，在测试集上效果一般，underfiting指的是在训练集测试集上效果都不好
    - 可以通过模型的评估指标在训练集测试集上的表现，对过拟合欠拟合进行判断
    - underfiting缓解方法：
      ①增加可用特征
      ②使用更加复杂的模型
      ③减少正则化系数
    - overfiting的缓解方法
      ①增加数据
      ②增加正则化系数
      ③调节模型参数，使之更加简单
      ④对于树模型可以对叶节点，树深度，节点分类最小值，shrinkage等进行调节
      ⑤对数据进行随机采样提高随机性
      ⑥对于深度学习：可以对数据进行一系列变换平移旋转调节锐化等等增加样本，earlystopping，增加噪声，Dropout，bagging/boosting,贝叶斯方法(说白了就是平均化/平滑)
20. 如何处理样本不均衡?
    - 首先对于样本不均衡数据不能使用acc的评估标准，可以结合业务场景使用AUC或者Fßscore。
    - 对于正负样本数据量都很大的时候可以进行下采样，可以使用EasyEnsemble策略
    - 对于正负样本数据量不大的时候可以使用上采样增加样本，使用SMOTE策略
21. xgboost有哪些参数会影响模型的复杂度？影响是怎样的?
    - eta，学习率，小一点不容易过拟合(shrinkage)
    - max_depth，树深度，越大越容易过拟合
    - min_child_weight，越大越容易欠拟合
    - max_leaf_nodes，越大越容易过拟合
    - gamma，分裂损失最小值，越大越容易欠拟合
    - alpha，L1正则化系数，越大越容易欠拟合
    - lambda，L2正则化系数，越大越容易欠拟合
22. xgboost的并行化体现在哪？xgboost的多分类如何做？xgboost的近似算法怎么做的?
    - 并行化体现在特征并行化。
    - xgboost多分类使用one vs rest方法
    - 近似算法体现在特征值的选取上，使用了近似直方图算法生成候选的分割点
23. 对Naive bayes进行解释?
    贝叶斯公式+条件独立+平滑，常用的平滑是拉普拉斯平滑，在NLP中词频统计的时候伯努利模型和多项式模型的平滑方式不同。
24. 有哪些对数据的基本处理方式
    - 异常数据的踢除
      具体要看使用的模型，对不同模型影响不一样
    - 数据的缺失值处理
      可以缺失值填充，可以删除
    - 定类数据的独热编码处理
      如果数据特征维度太高可以考虑哈希编码
    - 连续数据的离散化处理
      类别区间更重要
    - 连续数据的scaling
      使用GD优化算法的需要这步
    - 连续数据的分布变换
      box–cox，log变换之类
    - 日期型数据处理
      提取关键特征
    - 文本数据的正则化提取
25. LR与树模型差别，各自优缺点?
    - 差别
      ①树模型是非线性模型，LR是线性模型(当然可以对特征进行变换转化成非线性，这个暂且不论)
      ②LR需要对特征scaling，树模型可能不用
      ③LR对异常值敏感，树模型不敏感因为只是判断大小不是使用绝对数值进行计算
      ④基于算法原理，LR对数据整体把握较好，树模型对局部数据把握较好
    - 使用
      ①特征特别多且稀疏的时候使用LR比较好因为可以省时间，节省计算量
      ②LR对于分类特征无能为力
      ③树模型解释性比较好(不对，因为可以使用离散化onehot在LR上训练解释性也可以的)
26. 各种模型的feature importance的度量
    - 不基于模型的
      特征房差较大的特征度较大(熵高，信息量大)
      与label进行相关系数计算，相关系数越大重要度越高
    - 线性模型/LR
      根据得到模型的权重绝对值大小对特征重要度进行表征
    - Naive bayes
      可以通过对相同的词在不同类别中的频率之比取log作为重要度衡量
    - RF/GDBT
      ①每个特征在每棵树上做了多大贡献再取平均值，比较特征之间贡献度，主要使用的是基尼系数，仔细来说是对所有包含此特征节点的树都计算某个特征在某棵树上分叉前后基尼系数的变化，之后进行归一化处理，作为特征重要度的指标
      ②袋外数据错误率评估，数据自变量值发生轻微的扰动之后正确率与分类前正确率的平均减少量，计算方法类似于基尼系数
    - xgboost
      ①weight，某个特征在所有树中用于分裂样本个的数
      ②gain，某个特征在所有树中涉及到的所有节点的平均增益(指标是xgboost推导出的特有的公式)
      ③cover，这个是通过被分到该节点的样本的二阶导数之和，举个例子来说，某个特征作为节点(假设只有一个该特征节点)的对应分割样本数为10，那么此样本在这棵树的覆盖度就是10。说得通俗一点就是该特征所涉及到的观测值的相对数量情况(MSE)。
27. feature selection方式
    - 不基于模型的
      ①特征方差较大的特征度较大(熵高，信息量大)
      ②与label进行相关系数计算，相关系数越大重要度越高
      ③还有递归特征消除Recursive feature elimination(RFE)，基于特征重要度不断踢除重要度靠后的特征