技巧：

1、多看、多学、多实践

2、多看kernel, discussion

3、抱团取暖更好，切忌单打独斗

怎么打比赛？

1. 背景知识：数据分布
2. 查阅资料
3. 集思广义
4. 团队协作
5. 操作性极强：需要自己复现代码。

推荐资料：

《统计学习方法》

《利用Python进行数据分析》

sklearn、pandas、xgboost、lightgbm、numpy、ml-ensemble（mlens）官网

数据明确 -> 解决问题的思路

1. 数据一明确之后就要看到这是一个什么问题？（回归？分类？）
2. 这时候就要想起有哪些算法可以做回归？
3. 那么线性回归要什么样的数据才可以用上呢？
4. 数据中是否含有字符串或者缺失值？如何把它们变成数值型的？
5. 数据特征工程的思路，数据的EDA（pandas\_profling），特征选择，再就是特征组合或者特征分割等等。
6. 然后再就是算法的选择。

Why数据清洗？

对数据进行重新审查和校验的过程，目的在于删除重复信息、纠正存在的错误，并提供数据一致性。

难点：一般针对具体应用，因而难以归纳统一的方法和步骤，但是根据数据不同可以给出相应的数据清理方法。

数据清洗的方法：

1. 解决缺失值：平均值、最大值、最小值或更为复杂的概率估计代替缺失的值。
2. 去重：相等的记录合并为一条记录
3. 解决错误值：用统计分析的方法识别可能的错误值或异常值，如偏差分析、识别不遵守分布或回归方程的值，也可以用简单规则库（常识性规则、业务特定规则等）检查数据值，或使用不同属性间的约束、外部的数据来检测和清理数据
4. 解决数据的不一致性：比如数据是类别型或者次序型。

删除多列

更改数据类型

将分类变量转换为数字变量

检查缺失数据

删除列中的字符串

删除列中的空格

用字符串连接两列（带条件）

转换时间戳（从字符串到日期时间格式）

数据处理方法：对数变换、标准缩放、转换数据类型、独热编码、标签编码。

集成学习：Bagging，Boosting，Stacking（堆叠，做了交叉验证CV），Blending（融合，做了流出）

最大投票法：用于分类问题。

平均法：在回归问题中进行预测。

加权平均法：为所有模型分配不同的权重。

Stacking：经过预测得到模型的新特征，继续进行拟合得到新模型。

时间序列预测：根据过去时间里的数据对未来的时间里的target进行预测。

数据建模流程总结：

1. 读取数据，对train数据用sns.boxplot()进行可视化操作，观察是否存在异常值，若存在则删除。
2. 观察train数据，看是否有缺失值或不切实际的数值（如价格为负数）。若是数值型数据，则用中位数（中位数的计算需剔除缺失值或不切实际的数值之后再求）填充；若是字符型数据，则用众数填充。
3. 观察train和test数据是否有重复行，若有需删除重复行。
4. 让train和test数据集保持一致，再用pd.merge()将它们组合在一起。
5. 观察字符型数据，是否能提取出相应的信息（如从商店名字特征中提取出其所属的城市、商品的类型等），之后对提取出的信息进行LabelEncoder()转化成数值型的数据。如果是多文件的数据，需要根据组合数据的id特征进行merge。
6. 对于时间序列的预测，如对商品每个月的销售量预测，根据不同的月份、id或商店名等特征将销售量进行平移，构建lag特征。每次新作特征之后，就需和原来的数据进行合并。

计算机视觉：

基本任务：分类（表情识别）、检测（Bounding Box，以及里面内容的标记）、分割。

进阶任务：图像理解、视频理解、图像生成。

注意模型保存