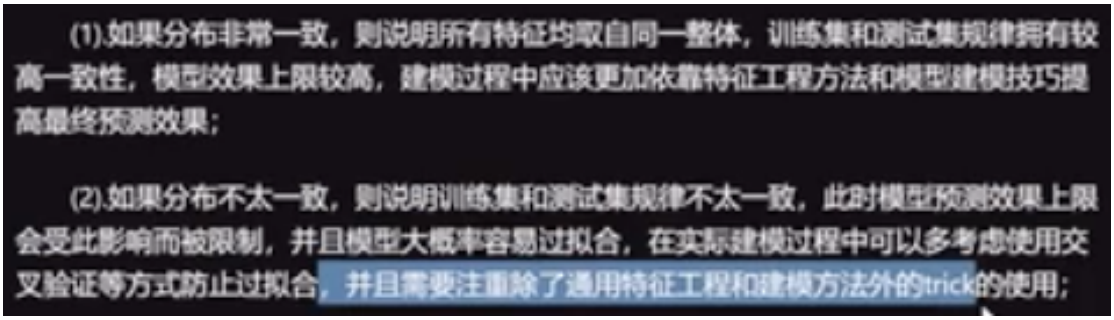
1 数据解读 **exploratory data analysis EDA**
2 数据预处理　数据清洗 data cleaning, data wrangling
3 数据可视化
4 划分 train, valid 数据集
5 特征工程
6 特征选择
7 特征优化
8 模型选择
9 训练模型
10 测试模型 提交结果

1 数据获取和解读(在 jupyter notebook 做 有助于复现。一般用到的工具是 pandas 和 SQL)
0）目的
familiarizing ourselves with the variables present in the data,
discovering potential hypotheses,
and identifying possible issues with the data.
0）文本文件建议从压缩文件读 可能更快，图片不建议 它本身就压缩了。
1）数据集整体概览
明确每个数据文件的内容含义，尤其是 train, test 数据集
分析不同数据集之间的关系，比如补充数据集和 train test 之间的关系。
2）查看数据集
检查数据文件的形状、大小是否正确。
查看数据集的前几个，知道数据结构、组成。
明白数据的每个特征的含义。
观察每个特征的数据类型，判断哪些可能有错 （data.dtypes

2 数据预处理　数据清洗
1）表格合并，表格格式整理
2) 错误数据的危害
如果数据集里有一些错误，好的模型仍然会收敛，但正确率会下降。如果把训练出来的模型部署到线上，新收集的数据可能会被影响 变得更差。
3）错误数据、异常特征的检测
3.1）对定性特征，可以按每个类别的出现次数排序，分析哪些可能是异常。
3.2) 对定量特征, 一般来说, 看最小值、最大值、平均值 来分析是否符合常理。data.describe()。
还可以通过 boxplot 找出 具体哪些数据是错误的。
还可以检查一些汇总项是否正确，比如求和项、特征计算项。
还可以检查个体信息是否独一无二 还是属于多种类别。

还可以判断 train test 数据是否符合同一规律，是否来自同一总体。方法：
单变量分布：分析 train test 两个数据集中，对应的特征数据的分布是否一致。
多变量联合分布：分析 train test 两个数据集中，对应的特征数据组合的分布是否一致。
作用：



3.3) 一般地，可以人工设置检测规则，比如说函数关系，逻辑判断关系、语法对应格式、语义对应格式（使用知识图）等。
4）观察错误值、缺失值的特点。
对于和缺失值相关的变量，通过分析它们的分布和它们与缺失值的关系，分析缺失的可能原因。
5）错误数据的处理
针对特定问题和目的进行分析，是否要丢弃 或补全这些缺失值。(可能用特殊数字标注，或用平均数填充)
可以通过 pandas 进行（一般不用），也可以通过多种专门的数据清理工具进行。一些常见处理思路如下：
5.1）丢弃不太重要的，缺失率比较多的特征
比如说缺失率>=30%

```
data.drop(columns=data.columns[null_sum > len(data) * 0.3], inplace=True)
```

5.2）过滤掉重要特征里 不符合实际情况的数据。比如说小于一平米的房子。
5.3）把错误分类的数据，手动分到合适的类。
5.4）把重要数据的格式转化成正确类型
比如说房价预测问题中，和价钱、税收有关的数据（李沐老师 实用机器学习）

```
currency = ['Sold Price', 'Listed Price', 'Tax assessed value', 'Annual tax amount']
for c in currency:
    data[c] = data[c].replace(
        r'[$,-]', '', regex=True).replace(
        r'^\s*$', np.nan, regex=True).astype(float)
```

和面积有关的数据

```
areas = ['Total interior livable area', 'Lot size']
for c in areas:
    acres = data[c].str.contains('Acres') == True
    col = data[c].replace(r'\b sqft\b|\b Acres\b|\b,\b','', regex=True).astype(float)
    col[acres] *= 43560
    data[c] = col
```

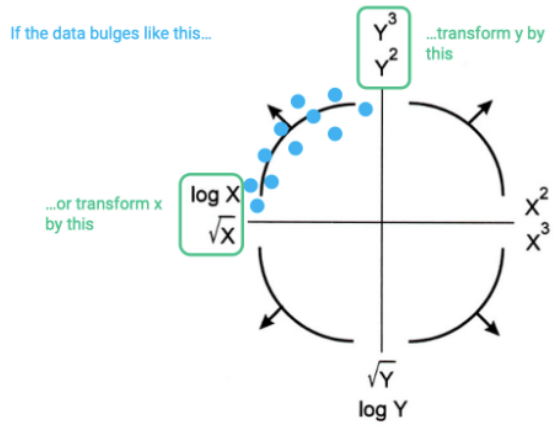
5.5）编码离散型变量 （名义型变量、有序变量）
5.6）无穷值处理

可能用最大取值代替无穷值
6) 使用 SQL 处理数据库里的数据
它是一种 declarative programming language，并且语句的顺序是固定的。

3 数据可视化（常用库是 matplotlib 和 seaborn）
1）对于可能相关的特征，分析它们的数据分布，推测可能的成因。
对于定性数据，可以用条形图分析它们的分布，也可以按每个类别的出现次数排序，分析哪些可能是异常。
对于定量数据，适合用柱状图（可找出 skewness 偏移）、box plot（可找出 outliers） 或 violin plot 分析分布，还可以用核密度估计来明确数据趋势（代表对数据分布概率的估计）。
2）找出所有可能和目标量相关的特征，分析它们之间的关系
对于**定性特征**和目标量的关系：可以按不同类别画出 box plot、核密度估计等，分析目标量在不同类下的分布特点，明确类别对目标量的影响。
对于**定量特征**和目标量的关系：散点图是最常用的方法。如果数据太多、太相似，可以调整点的大小和给数据增加小噪声（使用 jitter）来让图像更清晰。
其他的方法还有：lmplot, jointplot, hex plots, contour plots。
也可以画出不同量之间的 correlation maxtrix，比如：

```
_, ax = plt.subplots(figsize=(6,6))
columns = ['Sold Price', 'Listed Price', 'Annual tax
amount', 'Price per living sqft', 'Elementary School Score',
'High School Score']
sns.heatmap(data[columns].corr(),annot=True,cmap='RdYlGn',
ax=ax);
```

3）转化数据来发现潜藏关系
通常，数据之间的关系不是显而易见的，比如复杂的关系、异常值的影响、大量重叠数据的影响。把数据转化成另一种形式，可能让复杂的关系更清晰。
最常用的转化思路是，把数据转化成线性关系。在转化过程中，通过观察数据分布特点，不断思考能够让数据线性化的方式，直到完成转化。Tukey-Mosteller Bulge Diagram 可以帮助快速找到转化的思路。



4）使用 PCA 降维，有利于挑出重要的特征 进行分析。
5）为了展示多个维度的信息，可以用颜色、大小等不同图像特征来代表。

4 数据变换

1）实数数据归一化

归一化可以减小由于计量单位不同，造成的不同特征之间的过大差距（从而在正则化过程中，不同特征的系数的相对大小 可以反映特征对损失函数的影响大小。不会出现，由于特征数值大 造成系数过小 不能反映特征重要性的情况。）。也可以减小模型在梯度更新时的数据不稳定性。

常用的是 min-max normalization, z-score normalization, 不太常用的还有 decimal scaling, log scaling.

2）图片转化

常用的方法有下采样、裁剪、image whitening（注意：如果下采样程度比较大，且保存为中等质量的 jpeg 图片格式，可能会导致准确度有 1%的下降）

3）视频转化

视频数据先解码，再把它分割成小的视频片段，再从中采样部分图片序列。对压缩的视频数据，需要同时考虑存储和读取两个步骤的开销。对于采样出来的帧，可以用图片转化方法进行处理。

4）文字转化

语法和词根转化：把一个单词转化为一个常见的基础表达，比如 am, is ->be， car cars->car

词元化：把文字转化为一列词元。可能是一个个单词，一个个字符，也可能是一个个子词。（子词在大模型里比较常用，因为它可以用一个较小的字典来代表大量不同的单词信息）

5 特征工程

特征工程是 把原始特征转化为 更有意义的特征，用来帮助建模、提升模型性能。转化方式称为 feature function。

1）数值特征，可能直接使用，或转化成其他形式，或 binning 处理

2）对于时间特征，可能转化为 week of year, day of week 等特征。

3）对于非数值特征，可以用独热编码 转化为数值特征。

4）对于文字特征，可能通过 bag of words 模型 或 词嵌入 转化，也可能通过预训练模型。

5）对于图片视频，可能通过预训练模型抽取特征。

6）通用组合特征

通过统计不同离散特征在不同取值水平下、不同连续特征取值之和创建的特征。

7）业务统计特征创建

5 特征选择

1）filter 方法
借助皮尔逊相关系数，挑选和标签最相关的特征 进行建模
2）wrapper 方法
通过模型来筛选有效特征，把选出来的更重要的特征带入后续超参数优化和交叉验证过程
3）

6 特征优化

7 划分 train, valid 数据集
1）读取数据
1.1）读取 csv 文件
使用 pandas 的 read_csv 读取表格，并用 loc 方法将表格里的标签 和 图像像素值分离。
https://blog.csdn.net/Bigboss7/article/details/118597351
1.2)
Input type (long int) and bias type (float) should be the same
https://blog.csdn.net/Dr_maker/article/details/130603501
在读取数据的时候，将数据强制为 float32 格式

```
train = pd.read_csv(r"../input/train.csv",dtype = np.float32)
```

1）数据尺寸和格式转化
1.1）如果数据格式不适合直接 用模型处理，则先进行格式转化。比如说使用.view()调整数据的尺寸。
https://zhuanlan.zhihu.com/p/485688233
1.2）将数据变为可反向传播的 variable，记得要 import variable 函数
https://zhuanlan.zhihu.com/p/488010464
1.3)
Nll_loss 函数 通常需要标签是 long 类型。尝试把 target 转化为 long 类型。
https://deepinout.com/pytorch/pytorch-questions/279_pytorch_runtimeerror_nll_loss_forward_reduce_cuda_kernel_2d_index_not_implemented_for_int_pytorch.html

2）划分 train, valid 数据集，把数据转化为 tensor 形式
2.1）划分原因。
Valid 数据集作为在模型训练过程中 不可见的部分，可以评估模型对于未知数据的泛化能力，并依据此 来调整模型的复杂度，在 bias 和 variance 之间找到平衡。
2.1）分为 train, valid，使用 train_test_split

```
from sklearn.model_selection import train_test_split
```

用法如下链接 https://blog.csdn.net/weixin_48964486/article/details/122866347
2.2）使用 torch.from_numpy 把 numpy 数组转化为张量。

3) 数据打包和抛出
3.1）使用 TensorDataset 对 tensor 进行打包
TensorDataset 可以用来对 tensor 进行打包。该类通过每一个 tensor 的第一个维度进行索引。因此，该类中的 tensor 第一维度必须相等.
https://zhuanlan.zhihu.com/p/371516520
3.2）自定义 dataset 打包
理论解释 dataset 定义方法：
在 pytorch 中，如果需要自定义 Dataset，就需要实现__getitem__（）和__len__（）方法。这两个方法都是魔法方法。__getitem__（）用于取出一个序列当中的某一个，__len__（）用于返回序列的长度。
__getitem__()方法：
如果自己定义一个类，能否使用索引的方式获取这个类实例的属性值？答案就是使用__getitem__()方法。比如说，当 classname[2]这个语句出现时，就会触发__getitem__(self,idx)，这个方法就会返回 self.data[2]。
__len__（）
如果自定义一个类，是不是也能使用 len（）函数测量某个实例属性的长度呢？只要实现__len__（）就能实现这个功能。
理论解释 torch.dataloader 用法
这个方法要求输入的数据集是 `map-style datasets` 或 `iterable-style datasets`。以上定义方法就是将原来的数据集变为 `map-style datasets`。
`https://pytorch.org/docs/stable/data.html`
3.3）使用 dataloader 每次抛出一批数据。
https://zhuanlan.zhihu.com/p/371516520

4）图像增广
数据增强也叫数据扩增，意思是在不实质性的增加数据的情况下，让有限的数据产生等价于更多数据的价值。比如，阻止神经网络学习不相关的特征

8 模型选择和设计 (参考别人的做法)
1）单模型
2) 模型融合 综合多个模型的测试结果 作为提交的结果
2.1）voting 融合（均值融合）
2.2）加权融合
2.3）stacking 融合
3）定义 loss 计算方式
4) 合理调整模型的复杂度，在 bias 和 variance 之间平衡。比如说通过正则化
4.1）正则化 减小复杂度
通过限制特征 feature 的系数的取值大小，限制特征 feature 对模型的影响(尤其是，有的特征的系数会接近 0，此时该特征几乎不影响结果)，从而调整模型复杂度。常用的方法是 L1 和 L2 正则项。

L1 正则项，让参数绝对值尽可能地小。因此只有能有效减小损失函数的参数会保留，对损失函数影响不大的参数会尽量接近于 0 （在特征数值归一化后，不同特征的系数的相对大小 可以反映特征对损失函数的影响大小。不会出现，由于特征数值大 造成系数过小 不能反映特征重要性的情况。）。既保证了模型对数据的拟合效果，又控制了参数的复杂性。
L2 正则项，让参数的平方和尽可能地小。

9 模型调整
1）k-fold cross validation
目的：为了在多个不可见的数据集上 测试模型的性能，从而评估模型在新数据上的表现。
做法：把测试-验证集以 k 种不同的方式划分，并计算这 k 种方式上的平均验证误差，用来挑选合适的模型设置，比如说超参数。
2）超参数进行手动搜索和调优
从一个好的基线开始，比如说文献里报道的值。一次调整一个参数，看模型的效果变化。重复多次调整 从而得到一些 insight: 哪些参数对模型效果有重要影响，模型对参数变化的敏感度如何，什么是好的取值范围。
注意：保存训练日志和对应的超参数，便于对比、分享和重复。简单的方式是把文字用 log 保存，关键 metrics 用 excel 保存。更好的选择是，利用一些工具，比如说 tenesorboard 等
3）超参数进行算法自动搜索和调优
算法分为两种：黑盒、multi-fidelity.
黑盒方式里，对每个超参数选择 都会跑完训练过程。常见算法包括 网格搜索、随机搜索、贝叶斯优化器
Multi-fidelity 里，数据可能只是整个数据集的一部分、模型可能使用了更小尺寸的版本、训练过程可能提前终止，常见算法包括 successive halving, hyperband.
4）随机梯度下降
梯度下降 指的是，利用数据集的目标函数的负梯度方向 作为参赛更新的方向 。
随机指的是，每次取数据集的一部分 并计算梯度，用来估计全部数据的梯度，因此每个更新步骤 因采样的不同 会有随机性。它的优点是，计算比使用全部数据简单，而且经过较长时间也能收敛到最优解。
5）训练模型
3.1）定义训练函数
参考
https://www.kaggle.com/code/kanncaa1/pytorch-tutorial-for-deep-learning-lovers
https://blog.csdn.net/weixin_43845931/article/details/89166719
https://blog.csdn.net/m0_66785204/article/details/126926141
3.2) 结果处理
3.2.1）softmax
output = F.log_softmax(x, dim=1)
是对每一行的数据做归一化 并计算 log.
https://blog.csdn.net/m0_51004308/article/details/118001835
https://blog.csdn.net/qq_41375609/article/details/106078474
3.2.2) torch.max (output.data,1)[1]

第一个 1 代表 对每一行，得到最大值 和最大值的位置编号。
第二个 1 代表 取最大值的位置编号
https://pytorch.org/docs/stable/generated/torch.max.html
6）对于分类任务，可以做出 confusion matrix，通过 accuracy，precision 等指标来评价分类效果，也可以用 ROC curve 等评价分类器质量。

10 训练和使用模型
1）将所有的有标记的数据（包括验证集）作为训练集 训练模型。
训练和使用模型时，需要将所有的有标记的数据（包括验证集）作为训练集 训练模型。
2）用训练好的模型对测试数据进行预测，把预测结果和测试数据名称 都保存在列表中。
提交的文件格式 ImageId,Label
https://zhuanlan.zhihu.com/p/444076158
每计算完一批数据，就把结果添加到之前的预测结果后面。或者不用 dataloader 一批批加载数据，直接把整个测试数据集输入模型进行计算。
https://blog.csdn.net/tcn760/article/details/123867589
测试的时候，不应该随机从测试集里抽取。应该用比赛提供的测试集，按顺序运算 测试。
https://blog.csdn.net/weixin_40446557/article/details/103313708
3）把所有的结果写进 csv 文档保存。

11 提交 kaggle 项目结果