0813

1. 微积分和线性代数
2. 基本的Python能力（廖雪峰的Python教程，到面向对象编程一节即可）
3. 基本的Python数据处理能力（pandas和numpy）

pandas官方文档

Github上的pandas练习项目：<https://www.kesci.com>/home/project/5a8afe517f2d695222327e14

numpy官方文档

numpy菜鸟教程

常用机器学习算法：

* 线性模型原理和实现
* 集成学习模型（随机森林、GBDT、xgboost、lightgbm）原理和实现
* 深度学习模型、Tensorflow的使用
* 特征选择、模型调参与模型融合
* 不包含支持向量机（算法复杂度较高，在比较大的数据集上很难运行）

比赛平台：天池、Kaggle

腾讯广告赛、京东赛 – 数据量非常大、结果比较真实。

拍拍贷：https://ai.ppdai.com/mirror/show

微信公众号：麻婆豆腐AI

Github：https://iphysresearch.github.io/DataSciComp?sub=PF,AC,DM

比赛解决方案汇总Github：<https://github.com/Smilexuhc/Data-Competition-TopSolution>

机器学习的目标：以机器算法替代人工劳动，以自动地完成预测或决策。

机器学习：从大量的数据中学习规律，需要依赖于统计学工具。大部分机器学习问题都可以转化为预测问题。

计算机程序的逻辑仍然由人类编写和实现，但机器学习中的规律是由算法从大量数据中自动学习的。

线性回归：性能上的优越性、良好的可解释性。

系数b1：在其他条件相同的条件下，x1每增加1，y每增加b1。

练习1、使用sklearn估计孩子身高和父母身高之间的线性回归关系。

男女分开预测、正则化后再预测

计算机程序设计课程：<https://cn.udacity.com/course/design-of-computer-programs--cs212>

作业提交邮箱：[xmurpi@163.com](mailto:xmurpi@163.com)

0818

EDA

查看类分布：看类别分布是不是类别不平衡的。

查看特征的skew，对skew较大的特征取log。对一些值较大的特征也可取log可提升模型预测效果。

取log：保留特征原始的相对大小，但是差距变小。

pandas画图：<https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html>

数据透视表功能：<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.pivot_table.html>

时间特征处理：考虑时间的周期性，1天=24小时=24\*3600秒，划分为不同的hour, week, holiday, minute, day。

特征相关性分析：data.corr()，一般分析预测目标与哪个特征的相关性比较高。

若数据不平衡，其对相关性会存在很大的误导性。

正负数据平衡的比例：1:1最好，可容忍范围4:6。若类别数据比例低于3，则可认为数据不平衡问题。

异常值检测：

1. 画图
2. 四分位数法：IQR=Q\_3-Q\_1，Outliers>Q\_3+k\*IQR，或<Q\_1-k\*IQR，但主要找extreme outliers。

分类问题数据不平衡处理方法：

1. 收集更多数据
2. using weight（class\_weight属性）：样本不平衡时，模型会更倾向于去学习占比大的类别
3. 改变评价指标：混淆矩阵（多分类需转化为二分类），accuracy（正确率），precision（精确率），recall（召回率），F1 Score，AUC

Precision：预测为正样本中有多少是预测正确的。

Recall：原始有多少正样本被预测为正样本。

医学检测，更关注召回；法律判决，更关注精确。F1是精确和召回的调和平均（偏向于精确和召回更加均衡的模型）。

对于类别不平衡问题，一般以数量较少的一类作为正类，这样准确和召回更具有指示意义。

多分类问题下的precision和recall：

1. 采样

当正样本比例很低时（数据不平衡），Accuracy就失去了指示意义。

回归问题评价指标：MAE、MSE、Log Loss（神经网络）。

numba库：加速代码运行。

作业：

1. 根据不同权重class\_weight，画出LogisticRegression评价指标。
2. 补充完整评价指标函数定义(TP\FP\TN\FN)，思考fast\_auc()。
3. 复习所有东西。
4. 预习Resampling，Cross-validation。

0825 线性回归

参加DataFountain：离散制造过程中典型工件的质量符合率预测。

Q1：二分类问题不是在正态分布，难以用正态分布建模。 -> 用伯努利分布建模（抛硬币）

Q2：期望可能小于0或大于1，此时其含义难以解释。-> 引入映射函数（如Sigmoid函数，又叫逻辑函数）

为了获得统计上假设正确的模型（g(z)将映射到(0, 1)的函数），而从线性模型中引入逻辑回归模型。

线性回归：最小二乘法（以最小化数据集的均方误差为目标）。

对于线性回归方程：

最小二乘法试图找到能够使得下式：

取得最小值的一组。在最小二乘法中，一般把常数项也当成一个特征。

使用均方误差的原因：1）使用均方误差方便求导；2）能清晰的用统计学解释。

经验风险最小化：在训练集上做评价指标优化的方法。

梯度下降：一种贪心法，每次都向下降最快的方向走。其中，梯度是上升最快的方向，梯度的反方向即是下降最快的方向。在这个方向上走多远由学习率决定。

作业：自己实现线性回归的代码。

1、写课中的线性回归算法

选做：

1. 收敛条件的判断
2. 学习率的选择
3. 类的形式

0902 特征工程

<https://github.com/lambda-xmu/Data-Competition>

under-sampling: Tomek links

over-sampling: SMOTE

先通过SMOTE生成样本，再通过Tomek links降采样。

**k-fold Cross-validation**：把training data分成k份数据，迭代k次，将k-1份数据用于训练模型，对第k份进行预测。将k次迭代时第k份的全部预测与实际的标签进行比较，来评价模型。k次迭代时，每次均对testing data进行预测，最后的预测结果为k次预测结果的均值。

target encoding：对于类别特征col1，特征值为A占m次，其标签为1出现k次，其ctr为k/m。

对于稀疏类别，部分类别值出现次数很少，可考虑单独给一个均值的ctr值。

lightgbm能直接处理类别变量，不需要进行one-hot处理。

Label Encoding把出现次数最多的编码为0，依次编码。

Frequency Encoding：总共有n个样本，类别特征col1值为A的样本有m个，则编码为m/n。

leave one out：与计算ctr类似，只是每次计算时，需把自身从计数中删除，主要是为了防止过拟合。（对10天的数据，计算某一天数据的ctr时，不考虑该天的数值。）若数据集很大，如1亿个，删除一个效果不好，可随机删除1/4再做ctr。

特征选择：1）根据特征重要性排序，删除重要性低的；2）相关性。

连续特征分箱：将连续值分成n部分，即将连续变量转化为类别值。（将出现次数比较少的特征进行合并，统一为other）

组合特征：尽量组合有解释力的组合。

时间序列特征：1）时间窗（过去7天的min\max\mean\median\std，这样滑窗）；2）指示变量（是否是周六周日、是否是节假日）。

图特征;pagerank, graph embedding。

pyecharts

作业：将特征处理方式应用到比赛中。

0915 局部线性加权回归

局部加权线性回归：能更好的拟合非线性的关系。

比赛推荐：O2O优惠券使用预测 – 天池。

均方误差（MSE）：1）当离最优点很远时，收敛很快，越靠近最优点，梯度不断变小，有利于收敛。2）有清晰的统计学解释 – 极大似然估计。

平均绝对值误差（MAE）:梯度以恒定的速度前进。

1. 写出似然函数L（概率或概率密度的乘积）
2. 求似然函数的对数
3. 求对数似然函数的极大值点

伯努利分布的极大似然估计：

线性回归的极大似然估计：

1. 正态分布：
2. 观察N个样本，所有样本的取值均已知
3. 使用极大似然法估计

其最小化目标函数与线性回归中均方误差的目标函数一样。

Q：为什么线性模型选择均方误差作为优化目标？在正态分布的假设下，最小二乘法等价于极大似然估计。

中位数回归：以MAE为优化目标，预测目标为中位数。对异常值较为稳健。而均方误差对异常值比较敏感。

中位数回归的导函数不连续，不利于模型收敛。

为什么优化MAE得到的就是中位数，使用MSE得到的是均值？

当时，取得最小值。

分位数回归（Quantile Regression）：以Quantile Loss为优化目标。

1、中位数相当于50%分位数，而分位数回归可预测任意的分位数。

2、不对称的损失函数，使得对于预测值相比真实值偏大和偏小的惩罚程度不同。

3、分位数回归对于MAE左侧和右侧分别乘上“1-分位数”和“分位数”的权重。

对于10%分位数的回归，对于真实值比预测值小的部分，惩罚大；说明大部分的真实值大于预测值。

statsmodels.api.QuantReg

作业：使用statsmodels，预测身高数据集的8个10分位数。

Huber回归：以Huber Loss为优化目标 – 针对异常值比较多的情况。

Huber Loss综合上述两种损失函数，在一定范围内使用平方误差，超出后使用绝对值误差。

局部加权线性回归：使用线性回归对非线性关系建模。

原理：任何整体上非线性的关系，在局部上都可以用线性去近似。

没有提前给定线性模型，是在给出预测的点之后才去预测模型。

待预测样本与训练集中样本i的距离越大，权重越低。

决定了对于附近的点应赋予多大的权值。

缺点：1）在预测时仍然需要访问训练集，并且每次预测时都要重新估计回归方程；2）成本非常高，在大数据时代，甚至可能无法实现。

作业：基于sklearn的LinearRegression，实现一个局部加权线性回归模型（.py文件）。

1. 使用numpy计算待预测样本和训练集中样本点的距离，并转化为权重
2. 使用线性回归模型给出一个待预测样本的预测值
3. 使用循环给出每一个样本的预测值

github/drop-out 学习文档模式

0922 Entity Embedding

精确的错误（小样本）：根据中心极限定理，均值是服从正态分布的，通过线下随机取50组数据求平均，其值在均值的95%范围内，而线上却在剩下的5%范围内。线下非常精确，而线上预期完全不一样，使得线下没法模拟线上。

日期特征处理：提取年月日，在一年内的第几周。

可视化库：pyecharts

ECDF：主要使用于连续变量，表示所有贯彻样本中小于或等于某个值的样本所占的比例。

from statsmodels.distributions.empirical\_distributions import ECDF

sns.factorplot

sns,FacetGrid

训练模型的loss：log\_loss, auc, acc, mae, rms, mse

对于不常见的评价指标，需进行指标转换，使得loss等价于评价指标。

或者自己写损失函数。

embedding：使用更低维的向量来表示类别。

可以对类别标量做entity embedding，全部数据放进去训练。

转换成pickle格式，适合keras处理。

20190901 爬虫下载bz2，并解析出来

标签编码（Label Encoding）：适用于基于树的方法，但不适用于线性模型，因为其分配id的时候隐含了次序。

One-hot编码：如果类别多的话，会非常稀疏，难以达到最优；此外将各个类别视为彼此无关的独立实体。

1. 深度学习怎么处理列表数据呢？
2. 该怎么处理数据集中的类别变量？

Entity Embedding：使类别变量的每个类别都由向量表示，提供有关不同类别之间距离的信息。此外，训练好的Embeddings可以保存并用于非深度学习模型。

优点：已经学习的Embeddings可以被可视化以显示哪些类别彼此相似。

用途：

1. 在Embedding空间中查找最近邻，可以很好的根据用户的兴趣来进行推荐。
2. 可视化不同离散变量之间的关系。
3. 可作为监督学习模型的输入。

可视化方法：t-SNE

吴恩达：神经网络与深度学习

搭建图书推荐系统

CCF比赛的类别变量进行Embedding。

0929 模型验证与融合

blending：对多种模型的预测值直接取平均。1）等权融合；2）根据线下表现确定权重。

stacking：将多种模型的预测值作为特征输入，构建第二层模型再做交叉验证。

一般推荐使用K折交叉验证的out of fold prediction作为训练集第二层模型的特征，不要用训练集本身的特征。

思路1：建立Group级别的第二层模型，专门优化MAE损失。

1. 预测每一个工件四个类别的概率分布；
2. 按Group求平均，得到每个Group的平均类别分布比例；
3. 以每个Group的平均类别分布比例作为特征，预测Group中四个类别的真实比例。

假设两端概率相等，在中间随意移动预测点，不会影响期望MAE。