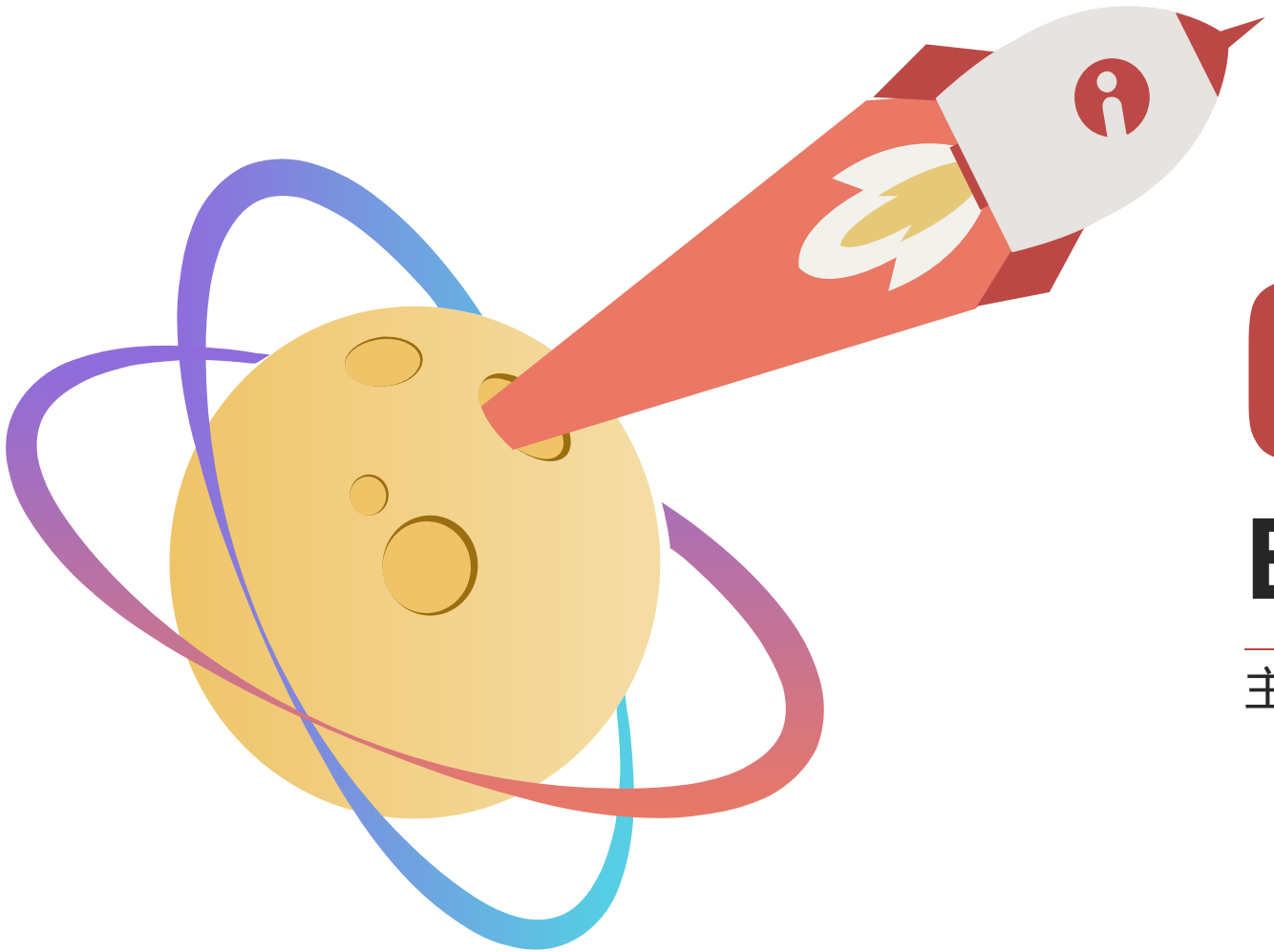
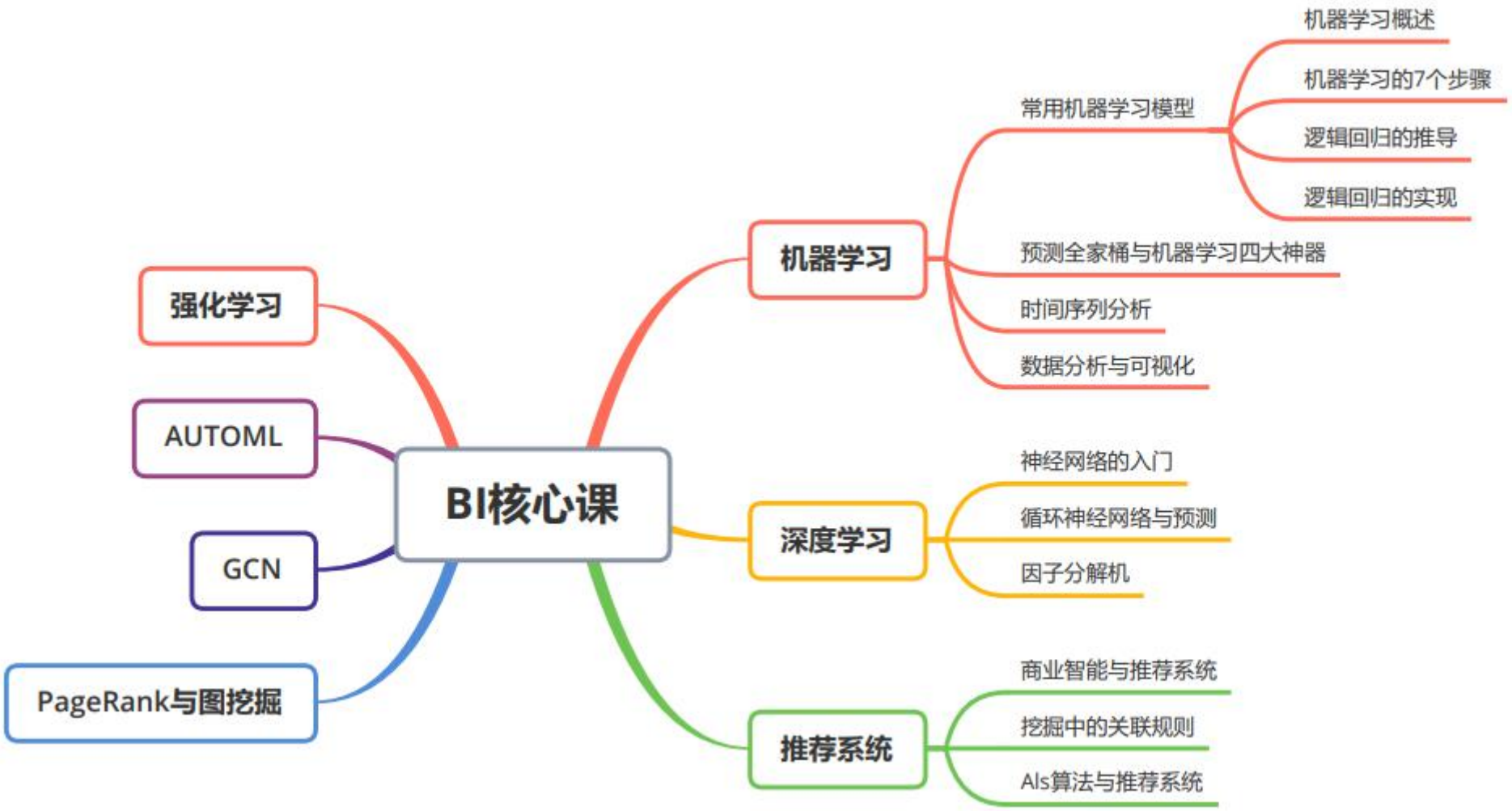


00

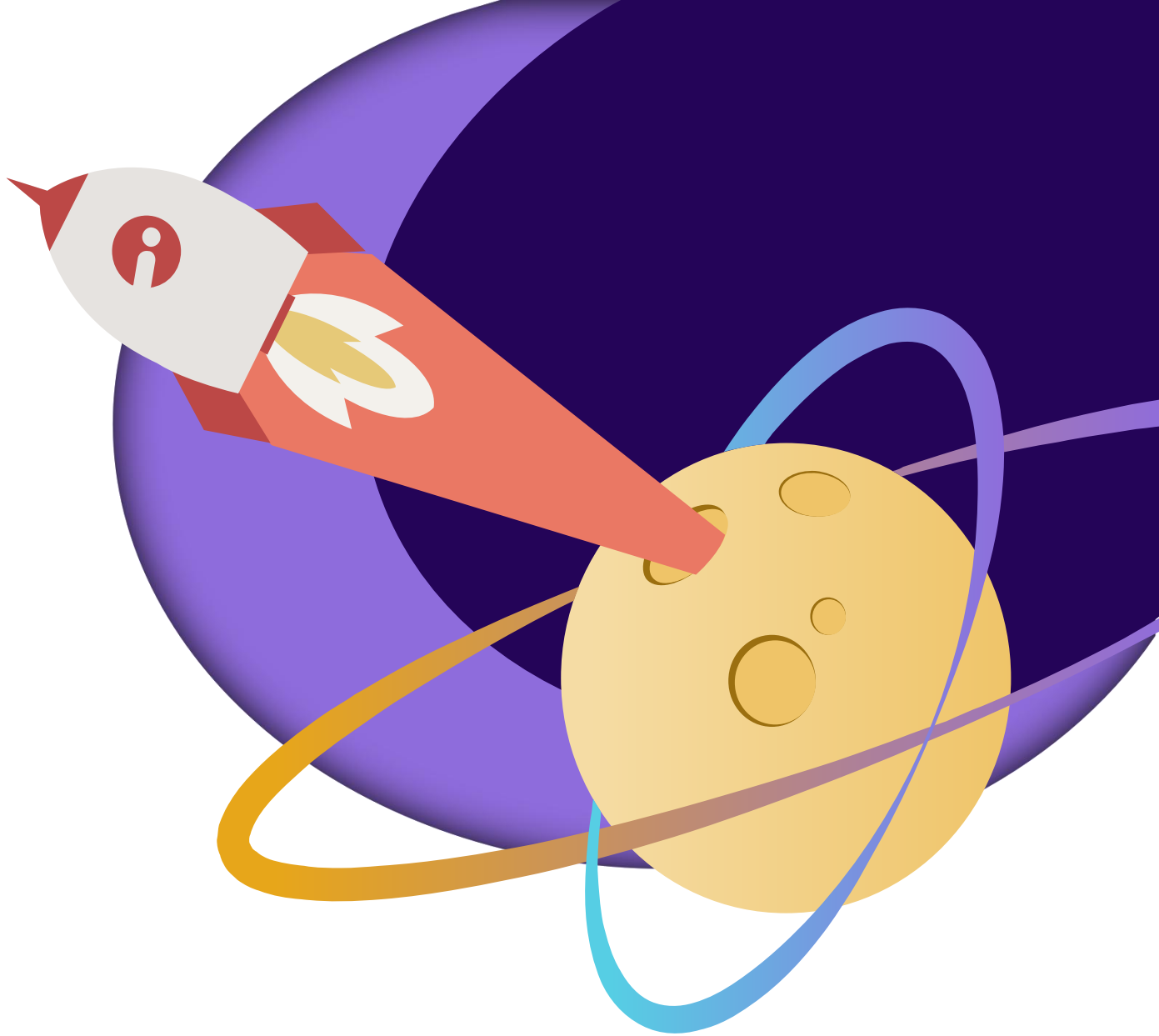
BI核心课大纲

主要介绍这门课的大纲

BI核心课大纲


BI核心课
机器学习
机器学习概述
机器学习的7个步骤
常用机器学习模型
逻辑回归的推导
逻辑回归的实现
预测全家桶与机器学习四大神器
时间序列分析
数据分析与可视化
深度学习
神经网络的入门
循环神经网络与预测
因子分解机
推荐系统
商业智能与推荐系统
挖掘中的关联规则
Als算法与推荐系统
强化学习
AUTOML
GCN
PageRank与图挖掘

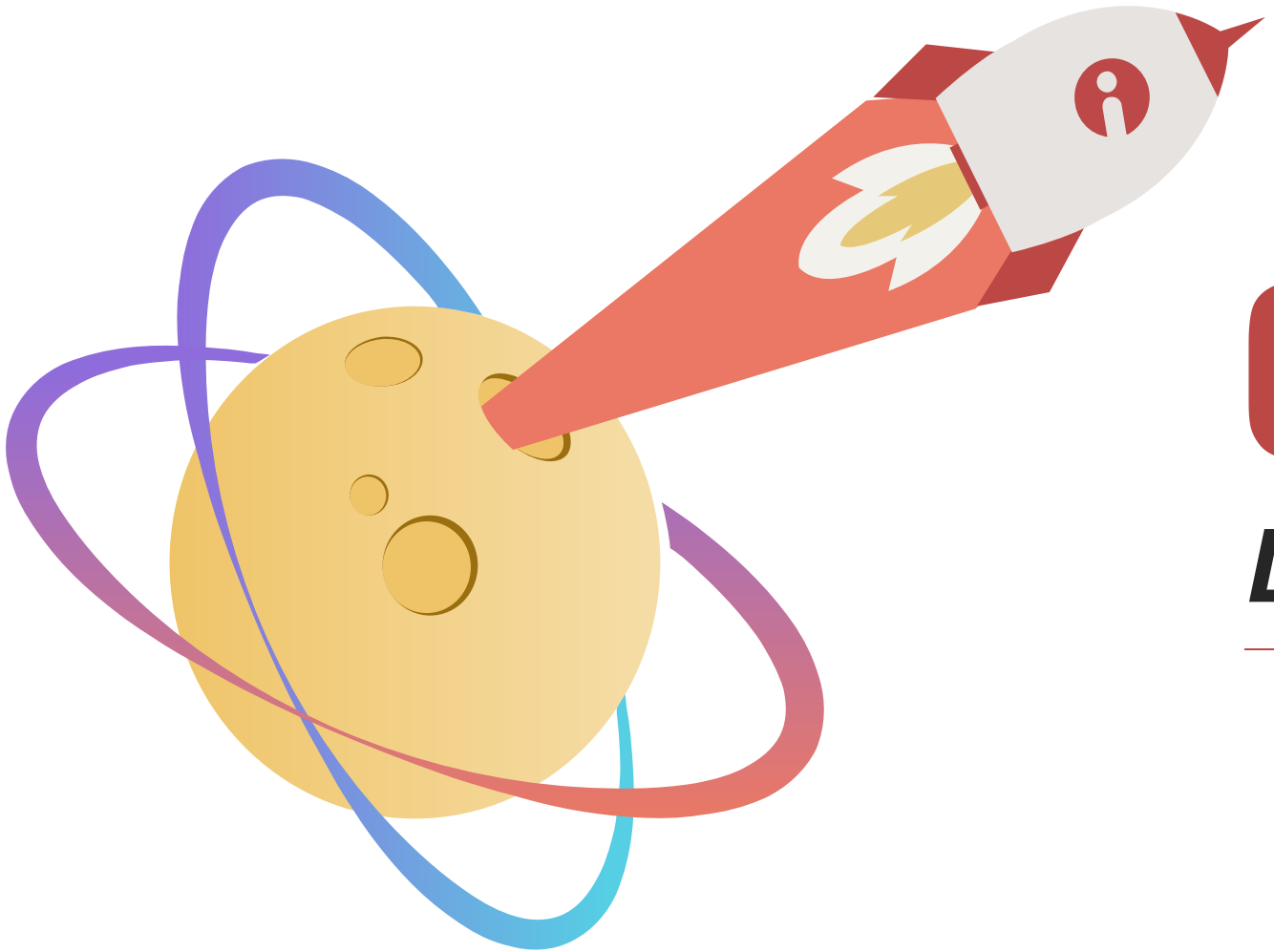
开课吧-BI核心课-PART1-机器学习



钟老师

2020.10

- *Lession1*学习目标
- 机器学习概述
- 机器学习的7个步骤
- 机器学习的实用工具
- 机器学习的预处理
- 逻辑回归



01

*Lession1*学习目标

Lession1学习目标

机器学习

机器学习概述

机器学习的步骤有哪些

机器学习步骤实践

使用传统机器学习完成初级项目

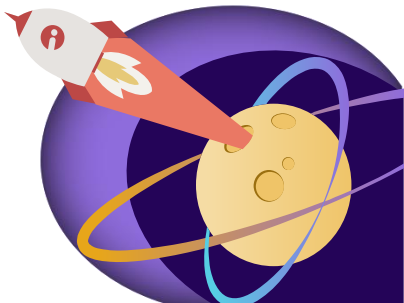
机器学习工具

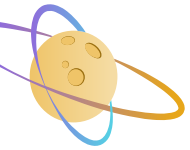
numpy简介

pandas简介

pandas merge groupby的用法

为什么pandas 和numpy 和机器学习





Lession1学习目标

逻辑回归

线性回归

逻辑回归简介

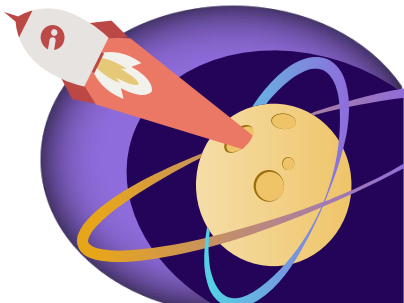
逻辑回归推导

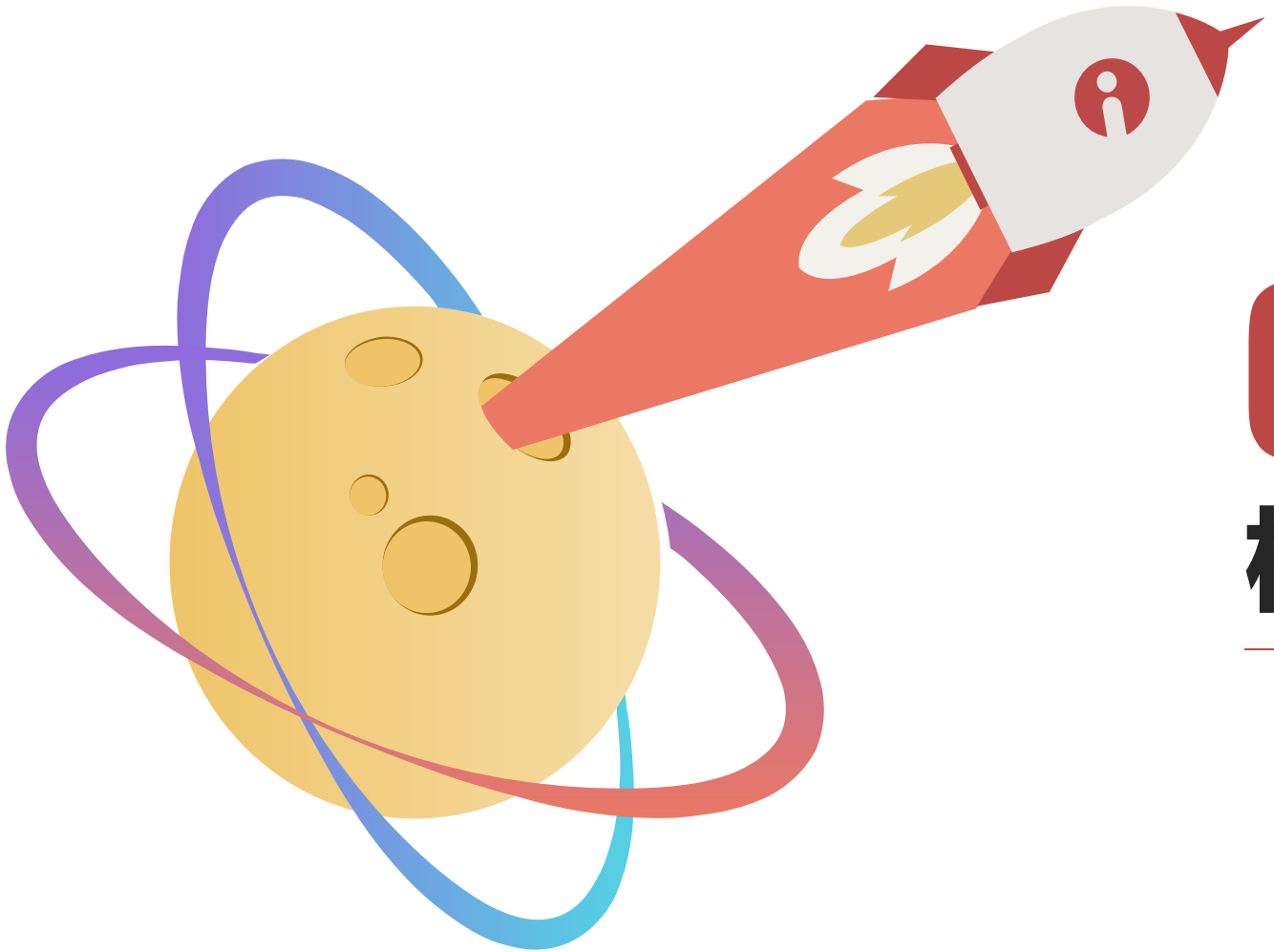
逻辑回归代码实现

Lession1实践

离职员工预测

课程学习方法Overview





02

机器学习概述

机器学习概述-人机对比

人类的学习

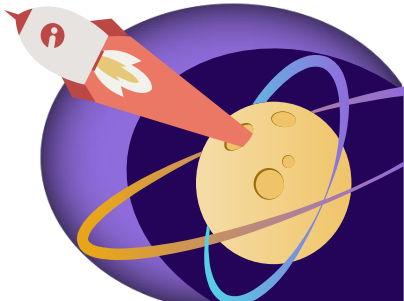
按逻辑顺序可分为三个阶段：输入，整合，输出。

人类往往是通过自己一定的基础知识(choose)，对某一件或者某些类别的事物(data)产生了一定的经验(training)，

然后在通过这些经验(model)，解决同类的事物或者相似的事物(predict),为了验证自己的是否错误，往往根据发生过的事情来对比(Validation)

机器学习

- 人类的不足：1)记性差;2)时间少;3)知识的爆炸
- 机器的优势：1)记得住;2)时间多;3)够内存
- 机器学习通过选择(choose)某些具有一定规律的算法，并利用现有的数据(data)，进行训练(training),并且通过训练得到一个模型(model),用于预测(predict)相同数据结构的数据.同时验证模型到准确性(Validation)



机器学习概述-应用场景

世界充满了大量的数据

2020年 世界的数据量44ZB

（

1ZB=2^10EB=2^20PB=2^30TB），

中国将达到8060EB（占比18%）

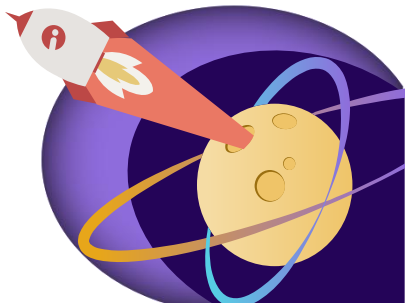
图片、视频，文字，语音，数字

人们也有大量的需求

预测：预测天气、股票、商品价格、企业发展、风险控制

“懂我”：推荐和“我”相关的item（商品、新闻、电影、音乐、用户、路径)

分类识别：某一领域的分类识别能力，图像识别、语音识别、自然语言处理

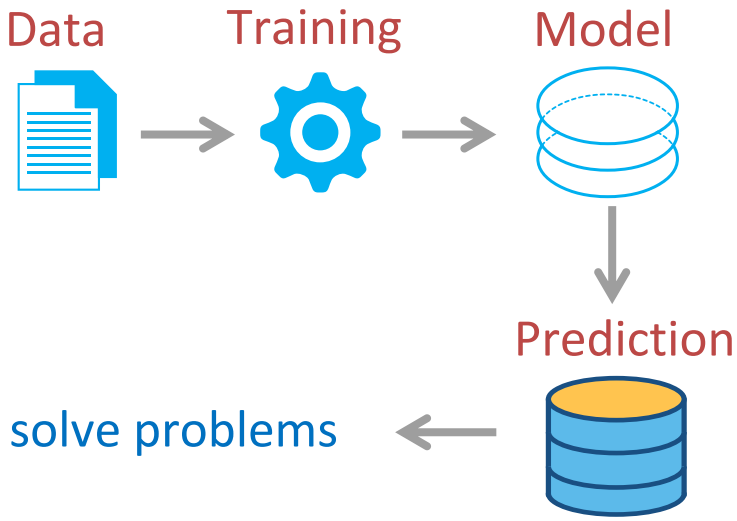


机器学习概述-机器学习的本质

机器学习就是利用数据，解决问题

Using data to solve problems

Training	Prediction
Using data	solve problems

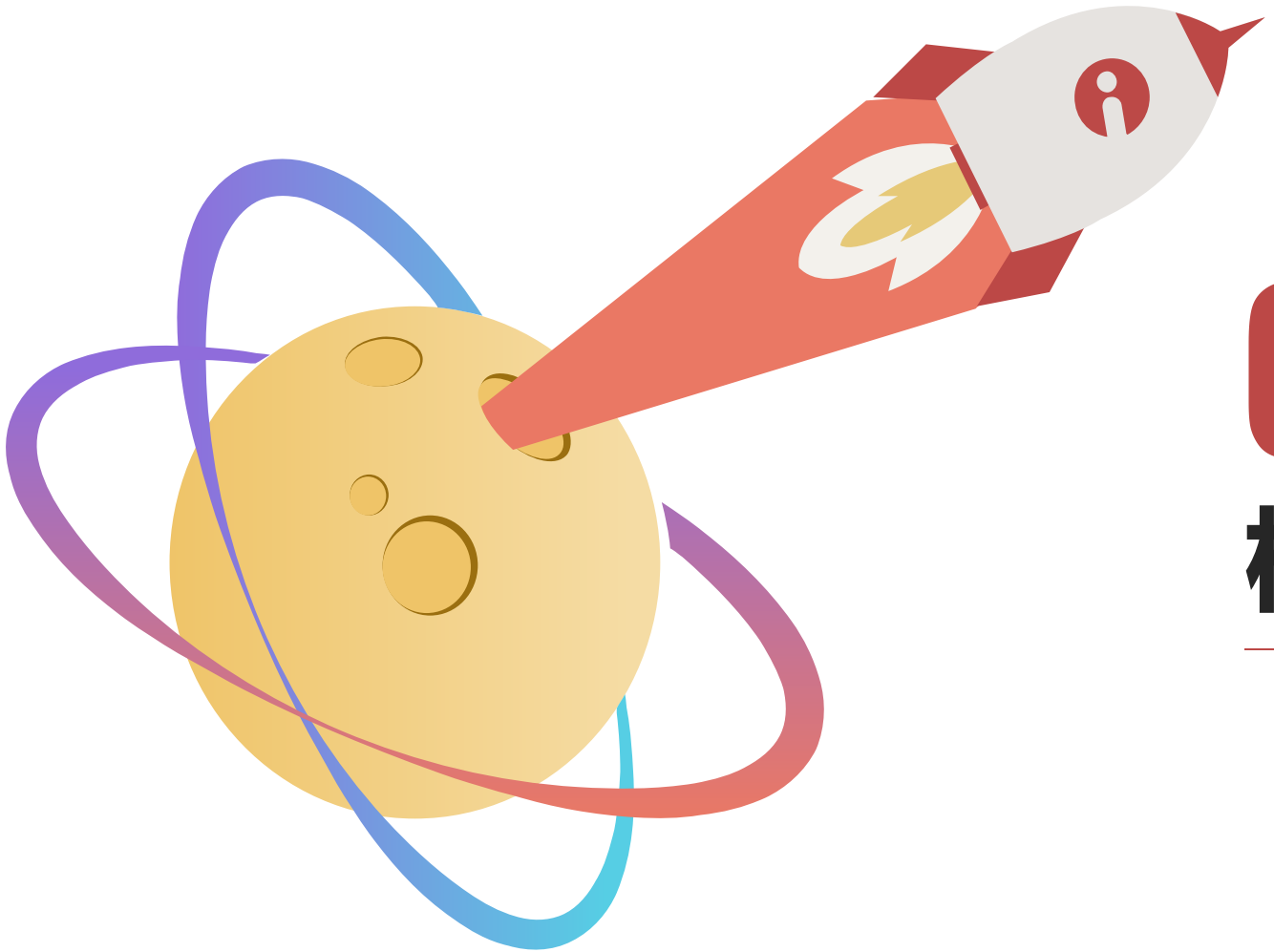


训练阶段：通过对数据的训练，创建一个预测模型并对其进行微调。

模型生成：预测模型可以从这些数据背后找出答案来，帮我们解决某个问题。

预测阶段：通过测试集完成模型评估，从而了解模型在测试集中的有效性。

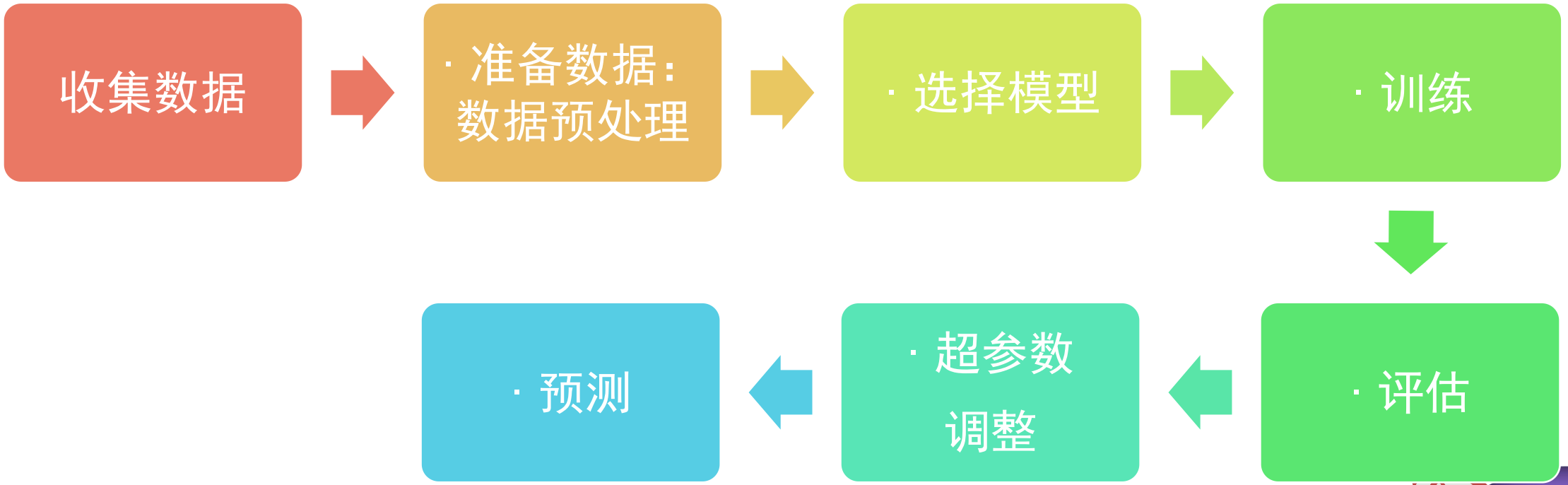
过程中，预测模型会被不断改进和使用。



03

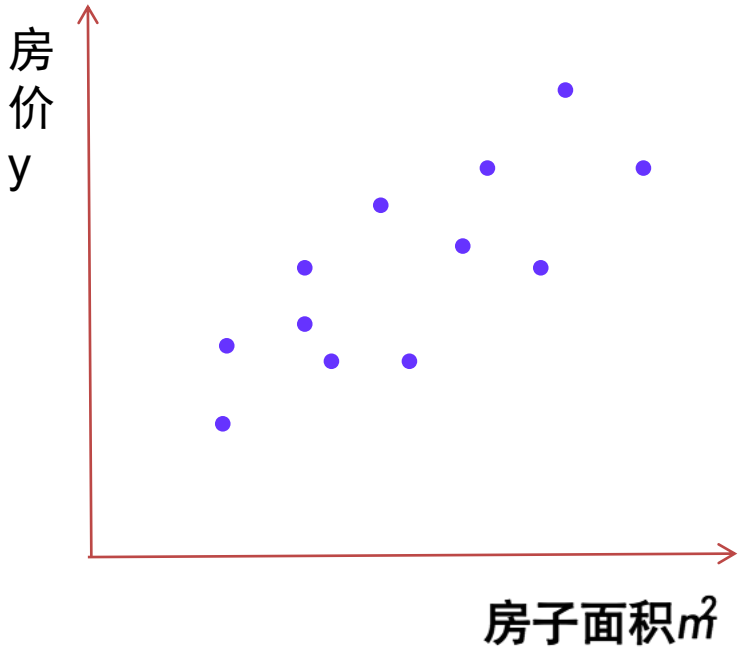
机器学习步骤

机器学习的7个步骤



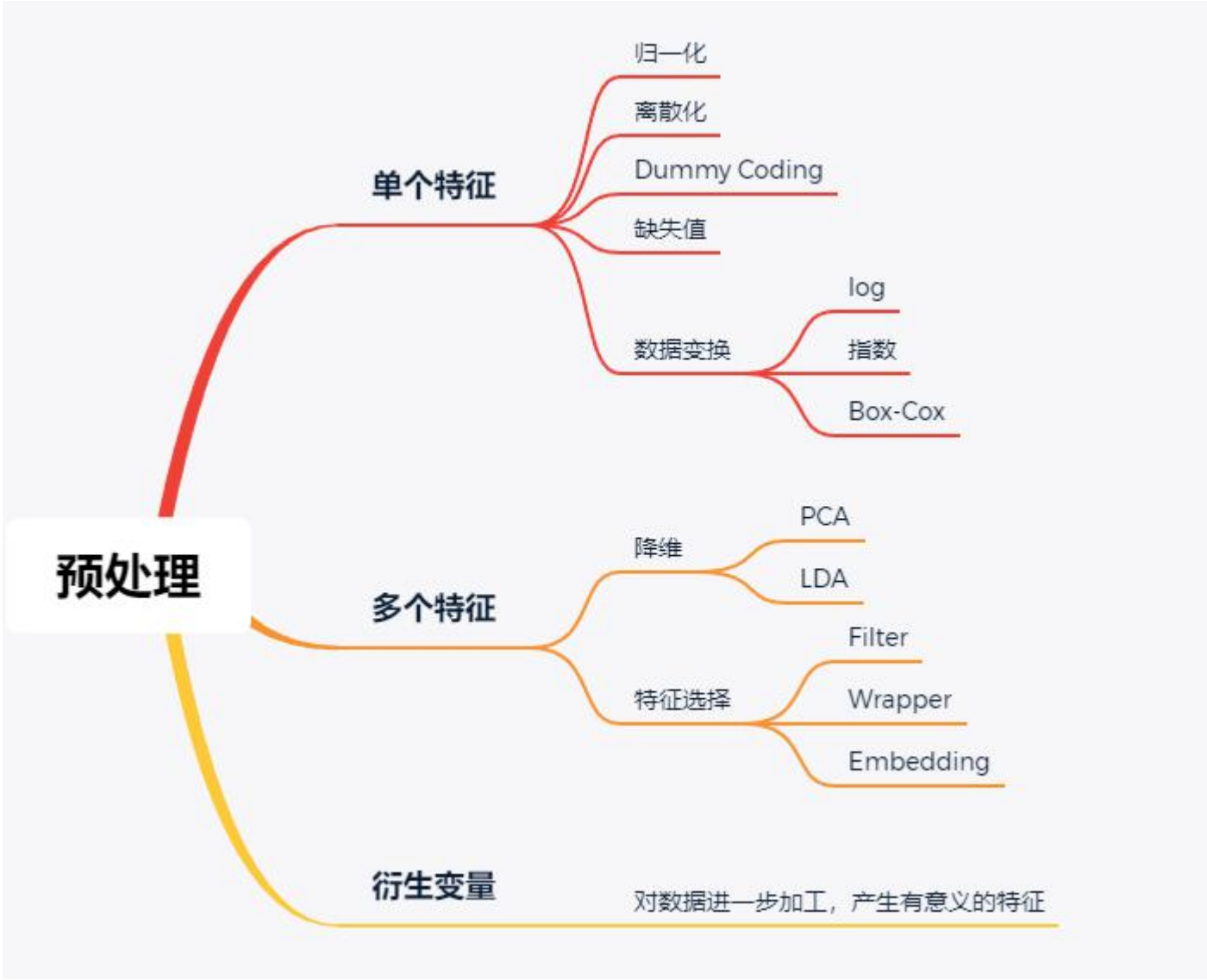
step1:收集数据

如何预测房价？



房子面积	房价
50	82
80	118
100	172
200	302
……	……

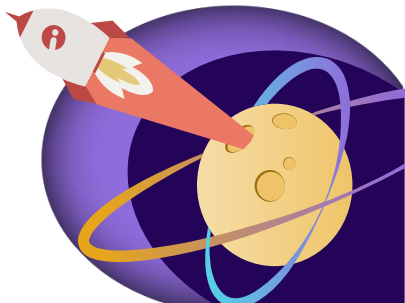
Step2:机器学习的预处理



step3:模型选择

什么是回归问题，什么是分类问题？

什么是线性回归，什么是逻辑回归？



step3:模型选择

如何判断杯子里盛的是水，还是饮料？

step3:模型选择

判断一个问题是分类，还是回归：

输出的数据类型：离散 or **连续**

线性回归

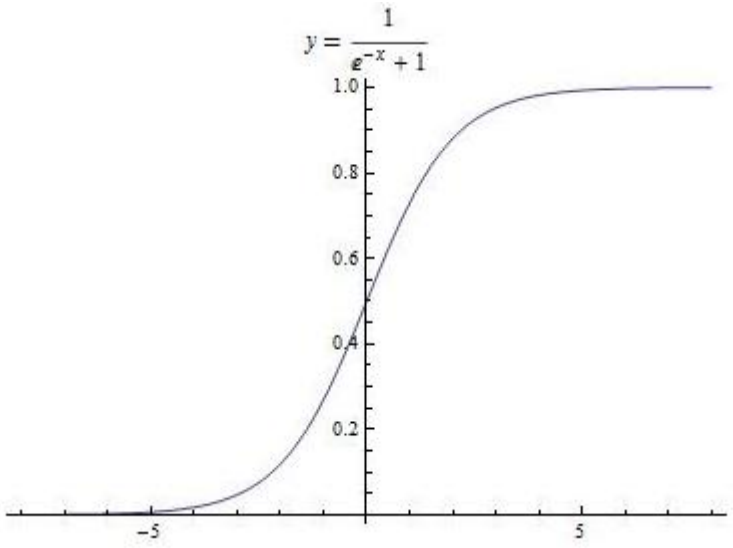
$$f(x) = w_1x_1 + w_2x_2 + \ldots + w_dx_d + b$$

$$f(x) = \mathbf{w}^T\mathbf{x} + b$$

逻辑回归

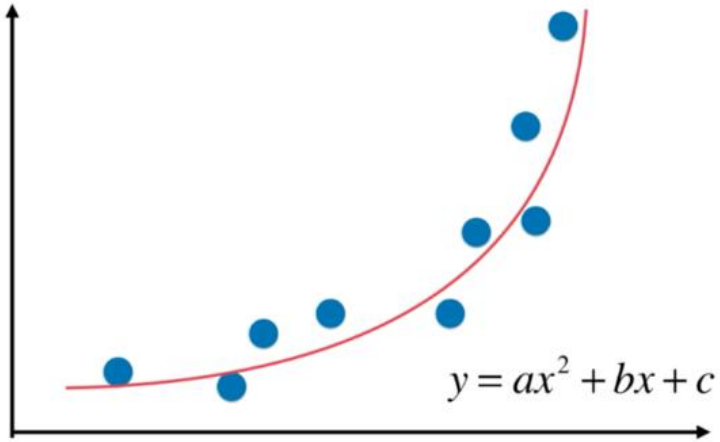
使用sigmod函数，实际上是分类算法

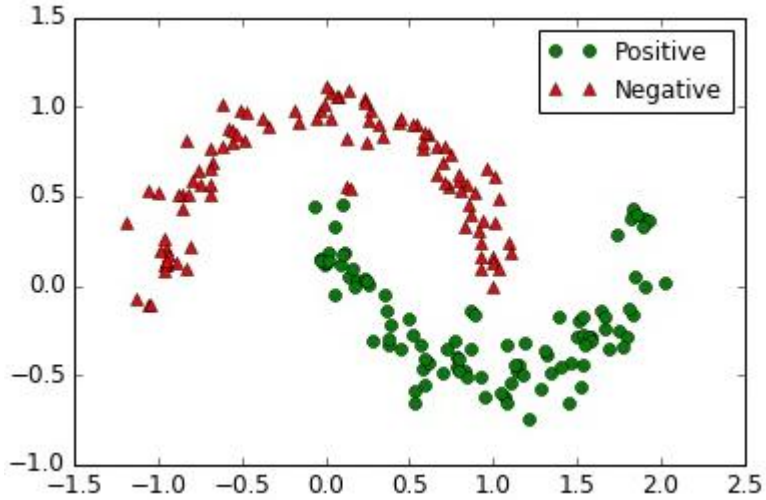
$$y=\frac{1}{1+e^{-(w^Tx+b)}}$$



$$y = \frac{1}{1 + e^{-(w^Tx+b)}}$$

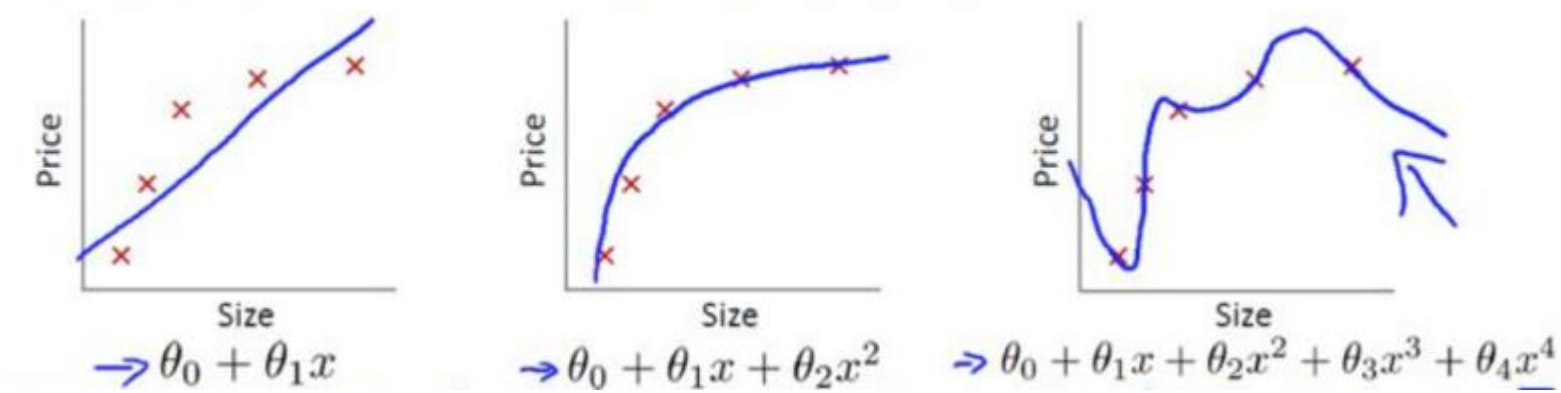
step3:模型选择


$y = ax^2 + bx + c$


Positive
Negative
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

step3:模型选择

如何用线性回归模型拟合非线性关系

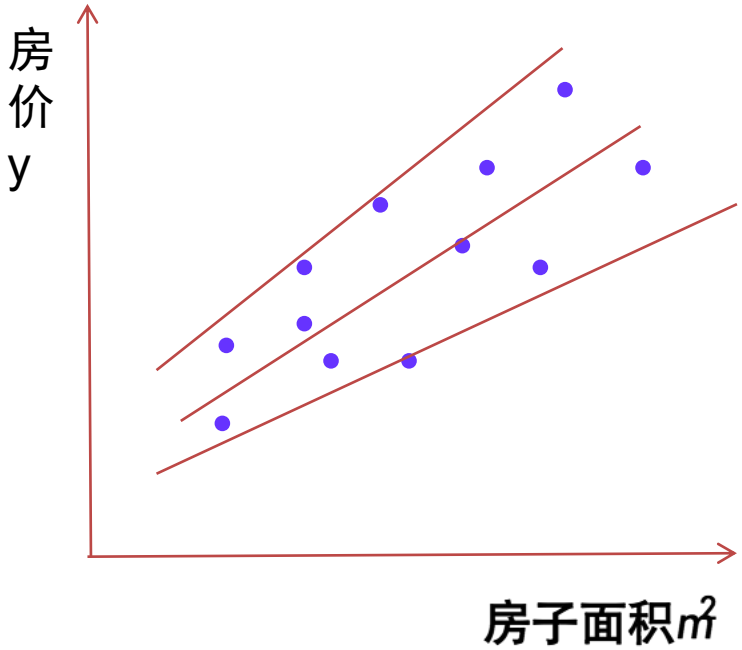


step3:模型选择

训练是机器学习的主要步骤

针对预测房价这个例子，我们可以采用简单的

线性模型：$y = w * x + b$



step4:训练过程

在机器学习中，我们有很多特征，基于这些特征，我们需要训练在Model中的权重w

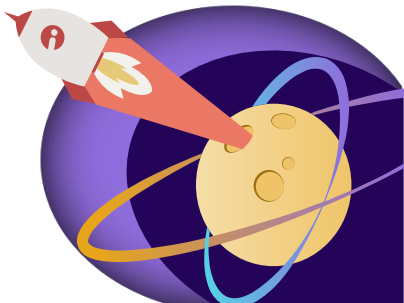
这些特征值构成的矩阵，称之为权重矩阵 weights

同时，还存在偏差，称之为 biases

房间大小	区域	周围绿化	周边配套	房型	房价y
50	海淀	A	A	style1	82
80	通州	B	A	style1	118
100	朝阳	C	B	style2	172
200	海淀	C	C	style3	302
……	……	……	……	……	……

$$f(x) = w_1x_1 + w_2x_2 + \ldots + w_dx_d + b$$

$$f(x) = \mathbf{w}^T\mathbf{x} + b$$



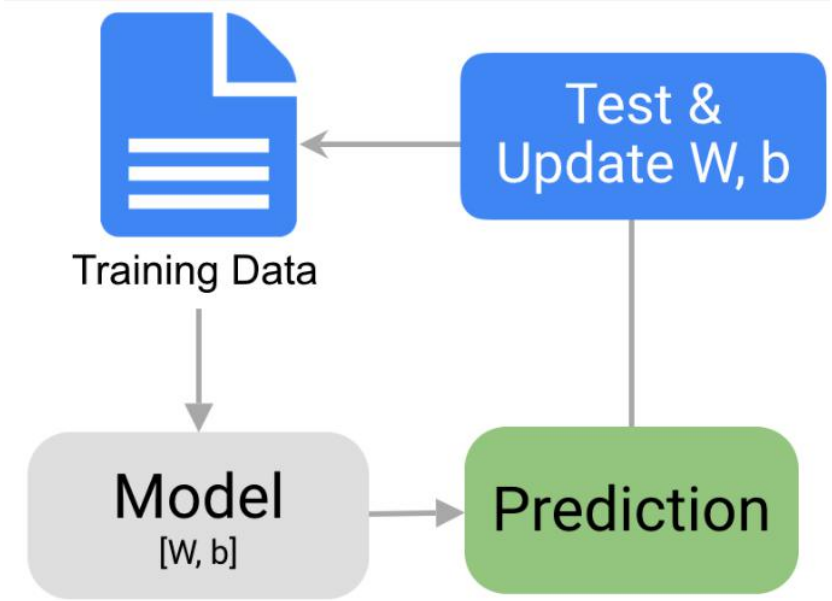
step4:训练过程

机器学习的过程，就是在搜索空间中对W和B进行搜索的过程，使得模型的准确率达到某个标准

一个训练步骤(training step)，称之为一次迭代。目的在于更新权重和变量

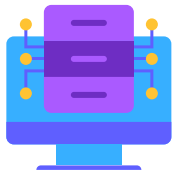
通过多次迭代，模型中的参数不断进行更新。就好像是在数据中进行线性拟合

当完成训练时，可以使用模型对房价进行预测

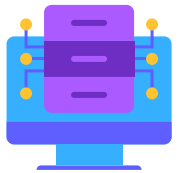


step5:机器学习的评估

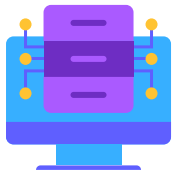
- 对数据的评估有多种方式：
- 我们通过会选择一部分数据作为验证集，比如20%或者10%，用于预测的数据，我们一般称为测试集。



train data



vali data

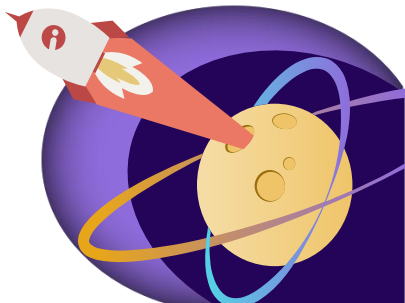


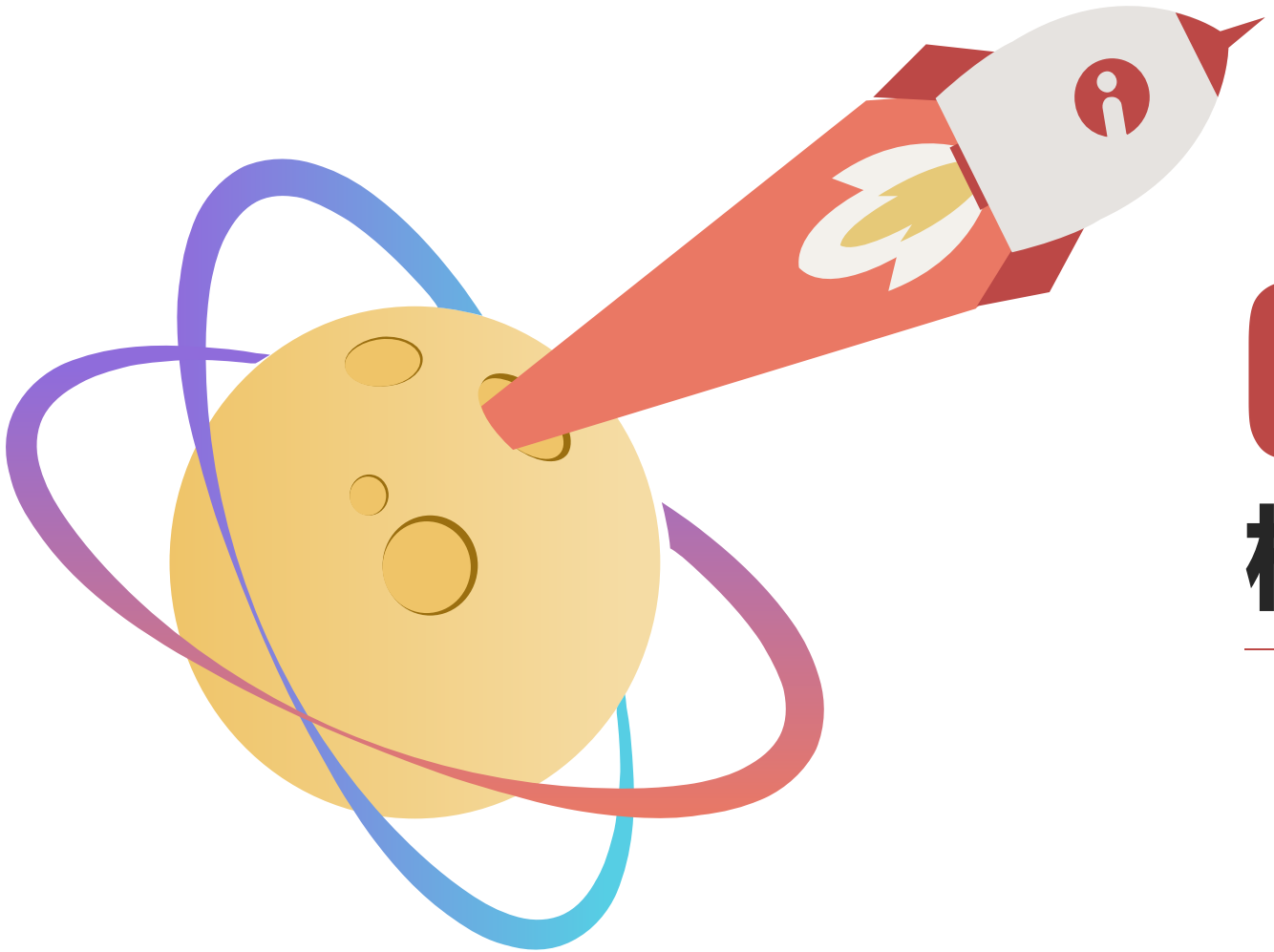
test data

step6:超参数调整

我们还可以对模型中的参数进行调整，比如epoch的次数，学习率等

这些参数通常被称为超参数。调整超参数的过程比起科学更像是艺术。这是实验性的过程，并很大程度上取决于具体的数据集、模型和训练过程

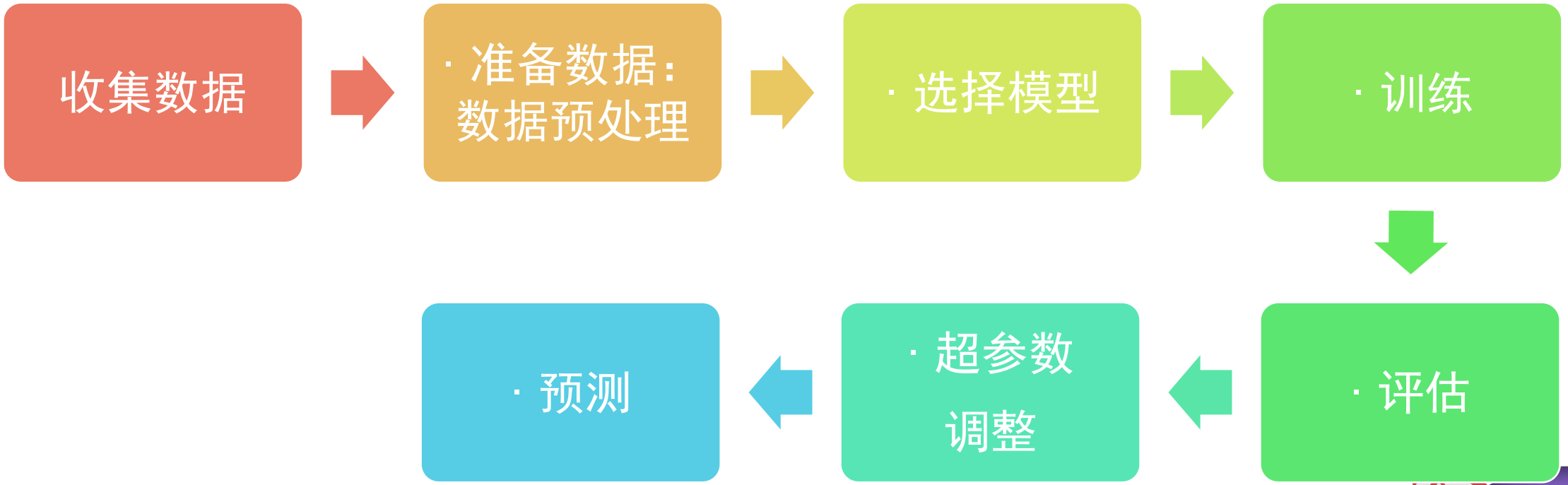




04

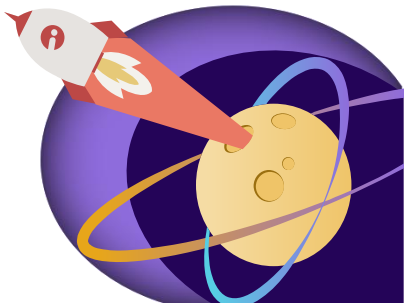
机器学习步骤实践

机器学习的7个步骤-实践



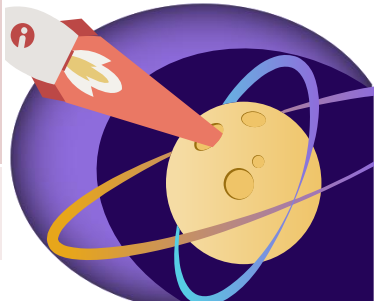
机器学习的7个步骤-实践

- Step1数据收集：http://archive.ics.uci.edu/ml/datasets/Bank+Marketing
- 数据收集-了解你的数据：
- 这些数据与葡萄牙一家银行机构的直接营销活动有关。营销活动是以电话为基础的。通常，同一客户需要多个联系人，以便了解产品（银行定期存款）是否被认购（“是”）或不被认购（“否”）（二分类）。
- 数据收集-字段介绍：



机器学习的7个步骤-实践

编号	字符名称	数据类型	字段描述	主要分
1	ID	Int	客户唯一标识	
2	age	Int	客户年龄	
3	job	String（类别）	客户的职业	categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown'
4	marital	String（类别）	婚姻状况	categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed
5	education	String（类别）	受教育程度	categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown'
6	default	String（类别）	是否有违约记录	(categorical: 'no','yes','unknown')



机器学习的7个步骤-实践

编号	字符名称	数据类型	字段描述	主要分
7	housing	String	是否有住房贷款	(categorical: 'no','yes','unknown')
8	loan	String	是否有个人贷款	(categorical: 'no','yes','unknown')
9	concat	String（类别）	与客户联系的沟通方式	(categorical: 'cellular','telephone')
10	month	String（类别）	最后一次联系的时间（月份）	categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed
11	day	String（类别）	最后一次联系的时间（具体几号）	
12	duration	Int	最后一次联系的交流时长	
13	campaign	Int	在本次活动中，与该客户交流过的次数	
14	pdays	Int	距离上次活动最后一次联系该客户，过去了多久（999表示没有联系过）	

机器学习的7个步骤-实践

编号	字符名称	数据类型	字段描述	主要分
15	previous	Int	在本次活动之前，与该客户交流过的次数	
16	poutcome	String	上一次活动的结果	(categorical: 'failure','nonexistent','success')
17	y	数值	就业变动率	(categorical: 'cellular','telephone')

机器学习的7个步骤-实践

Step2数据预处理：

读取数据

```
train = pd.read_csv(path+'input/train_set.csv')

test = pd.read_csv(path+'input/test_set.csv')
```

数据概览

```
train.describe()
```

test的y处理

```
test['y']=-1
```

数据合并

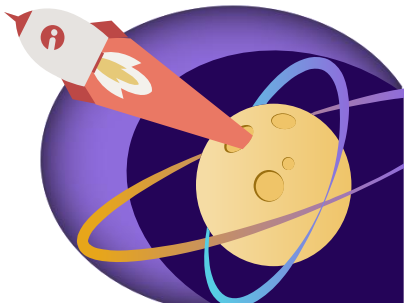
```
data=pd.concat([train,test])
```

筛选类别特征

```
cat_col = [i for i in data.select_dtypes(object).columns if i not in ['ID','y']]
```

Labelencode

```
for i in tqdm_notebook(cat_col):
    lbl = LabelEncoder()
    data[i] =
lbl.fit_transform(data[i].astype(str))
```



机器学习的7个步骤-实践

Step3选择模型：

选取特征

```
feats = [i for i in data.columns if i not in ['ID','y']]
```

选择模型

```
import lightgbm as lgb
model = lgb.LGBMClassifier(
        boosting_type="gbdt", num_leaves=30, reg_alpha=0, reg_lambda=0.,
    max_depth=-1, n_estimators=1500, objective='binary',metric= 'auc',
    subsample=0.95, colsample_bytree=0.7, subsample_freq=1,
    learning_rate=0.02, random_state=2017
    )
```

筛选训练集

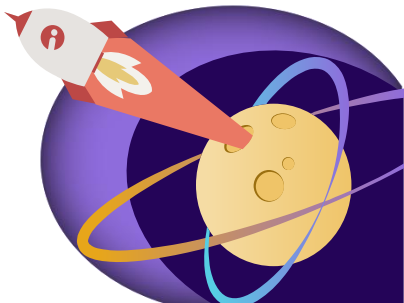
```
data1 =data[data['y']!=-1][feats]
```

筛选训练集lable

```
label1 =data[data['y']!=-1]['y']
```

测试集

```
testx= data[data['y']==-1][feats]
```

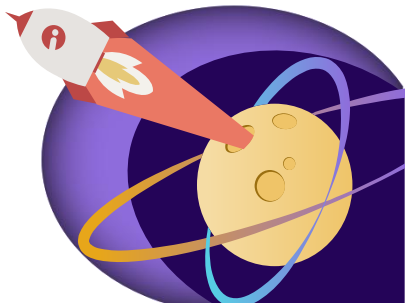


机器学习的7个步骤-实践

Step3选择模型：

筛选训练集
筛选验证集

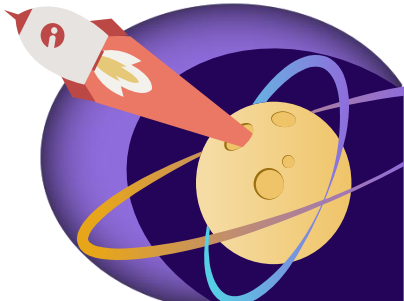
```
train_x, test_x, train_y, test_y = train_test_split(data1, label1, test_size=0.3,
random_state=42)
```



机器学习的7个步骤-实践

Step4-7模型训练-预测：

模型训练	model.fit(train_x,train_y)
模型评估	auc_score(y,y_predict)
超参数调整	model = lgb.LGBMClassifier(boosting_type="gbdt", num_leaves=30, reg_alpha=0, reg_lambda=0., max_depth=-1, n_estimators=1500, objective='binary',metric= 'auc', subsample=0.95, colsample_bytree=0.7, subsample_freq=1, learning_rate=0.02, random_state=2017)
预测	test_pre = model.predict_proba(testx)[:,1]
生成结果	pre=data[data['y']==-1][['ID']] pre['pred']=test_pre



例子2：使用LogisticRegression对MNIST手写数字进行识别

- MNIST 数据集是经典的手写数字识别数据集，每个样本28*28

- 精简版MNIST：一共1797个
- 完整版MNIST：一共6万个样本（5万个训练，1万个测试），每个样本28*28
- http://deeplearning.net/data/mnist/mnist.pkl.gz

例子2：使用LogisticRegression对MNIST手写数字进行识别

```
# 使用LR对手写数字分类

from sklearn.model_selection import train_test_split

from sklearn import preprocessing

from sklearn.metrics import accuracy_score

from sklearn.datasets import load_digits

from sklearn.svm import SVC
```

引用包

```
# 加载数据

digits = load_digits()

data = digits.data

 # 分割数据，将25%的数据作为测试集，其余作为训练集

train_x, test_x, train_y, test_y = train_test_split(data, digits.target, test_size=0.25,
random_state=33)
```

数据加载

```
# 采用Z-Score规范化

ss = preprocessing.StandardScaler()

train_ss_x = ss.fit_transform(train_x)

test_ss_x = ss.transform(test_x)
```

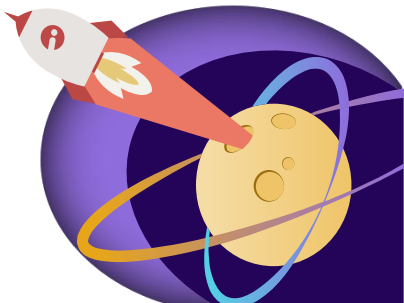
数据预处理

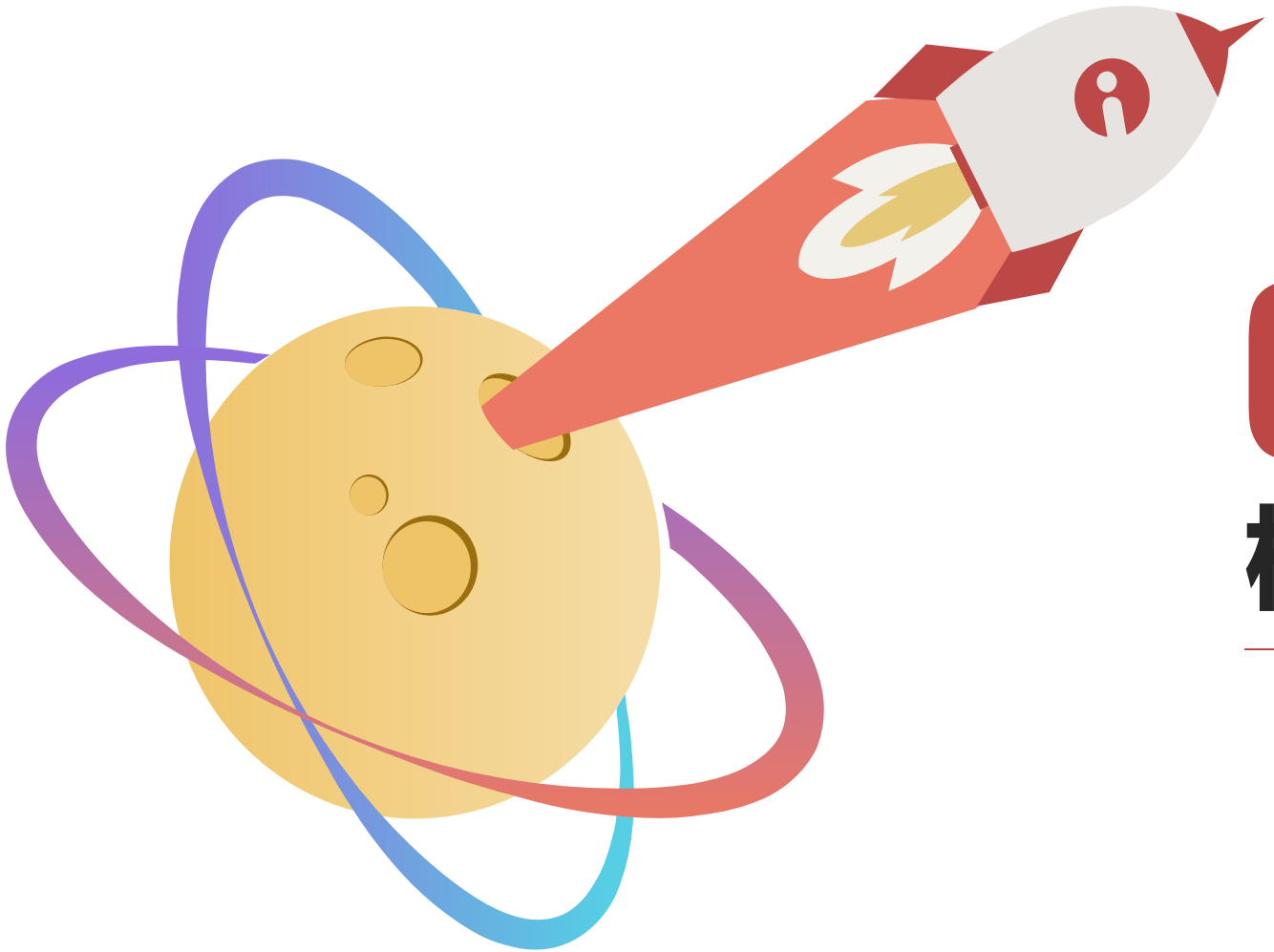
```
# 创建LR分类器

lr = LogisticRegression()

lr.fit(train_ss_x, train_y)
```

模型训练

```
predict_y=svm.predict(test_ss_x)

print('SVM准确率: %0.4lf' % accuracy_score(test_y, predict_y))
```

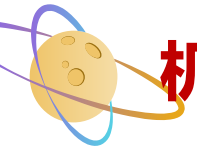
模型评估





05

机器学习工具

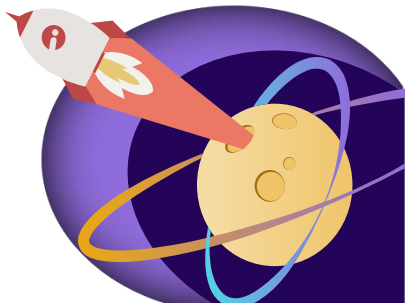


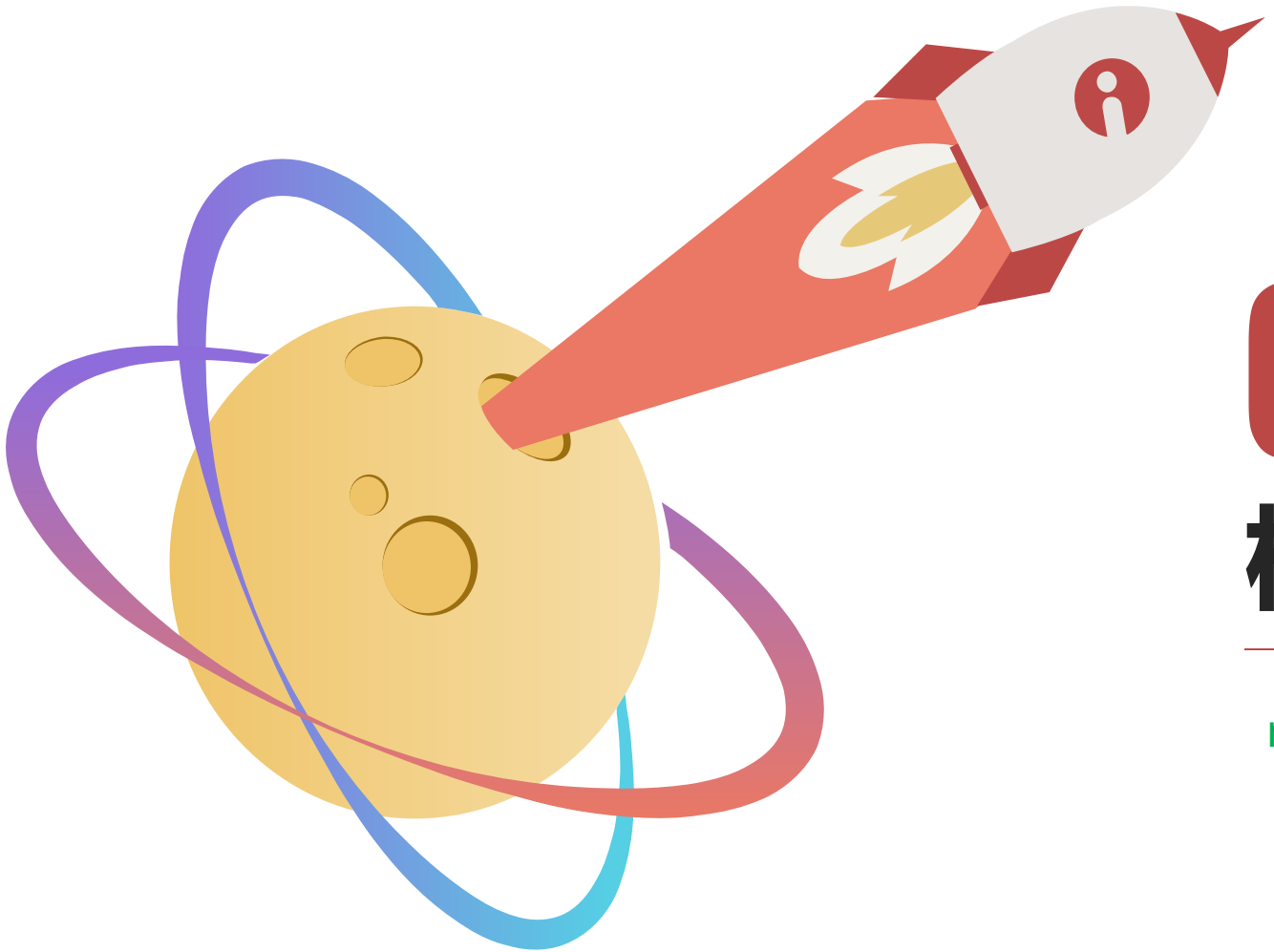
机器学习工具

有很多机器学习工具可供选择，课程主要使用Python，已经是数据分析的首选语言。

Python中的常用工具包：

- Numpy
- Pandas
- Sklearn

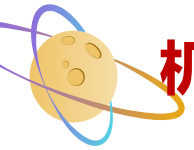




05

机器学习工具

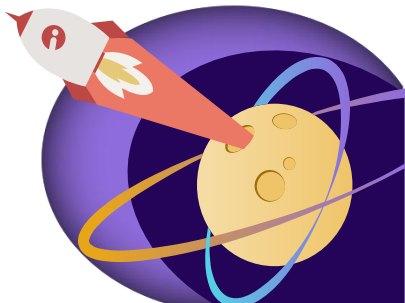
numpy

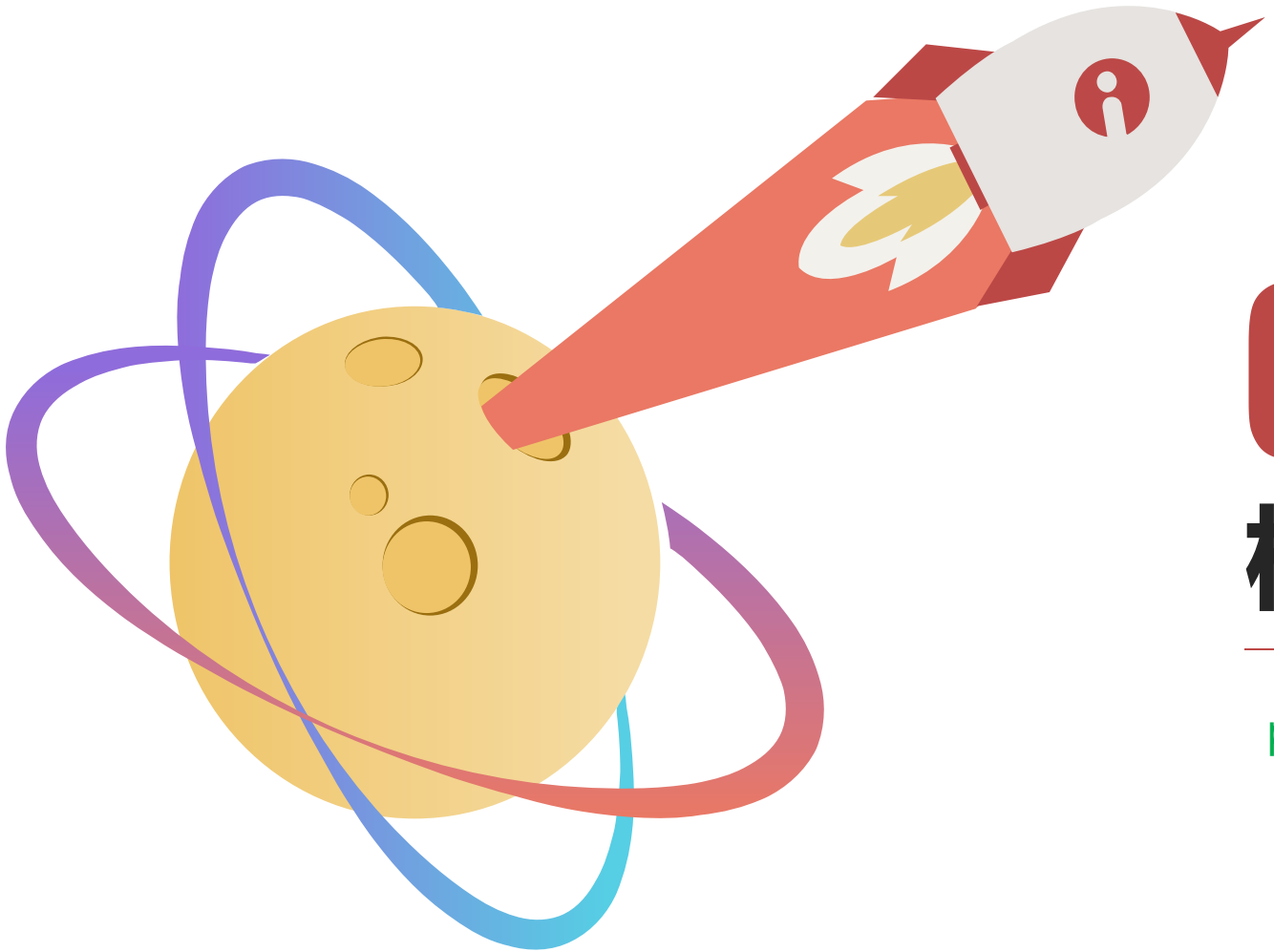


机器学习工具-Numpy

Python中的常用工具包：

- Numpy(Numerical Python)
- 1.数组、矩阵
- 2.包含线性代数、傅立叶变换、随机数等
- 3.学习numpy （基础）可以进一步学习pytorch/sklearn/tensorflow/keras等

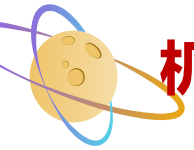




05

机器学习工具

pandas

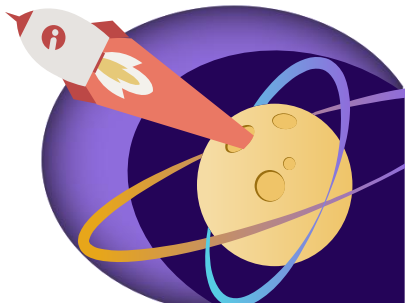


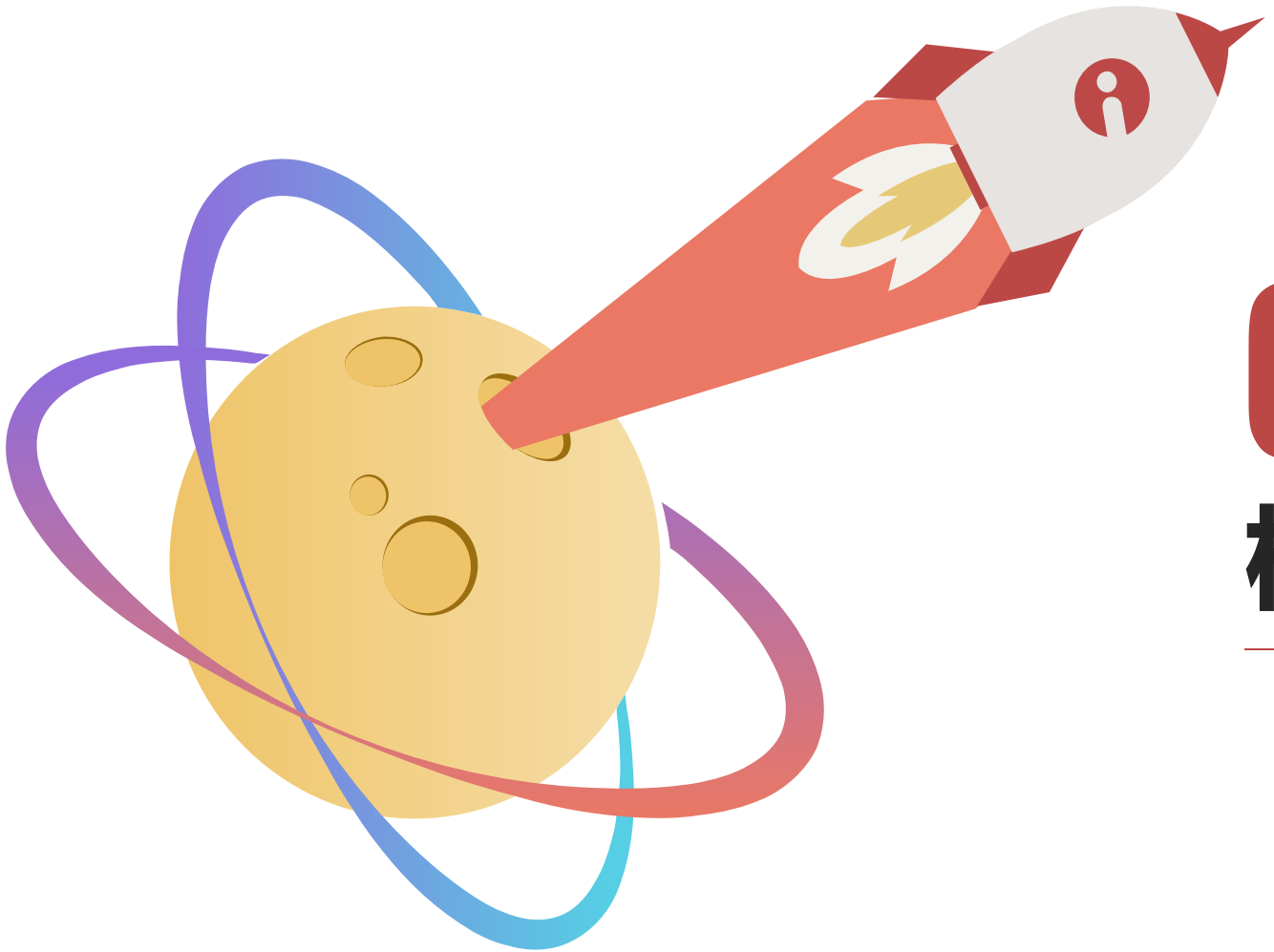
机器学习工具-Pandas

Python中的常用工具包：

- Pandas

基于numpy 的一种工具，为了解决数据分析的任务而创建，其中纳入了大量的库和一些标准数据，提供了大型数据所需的工具。

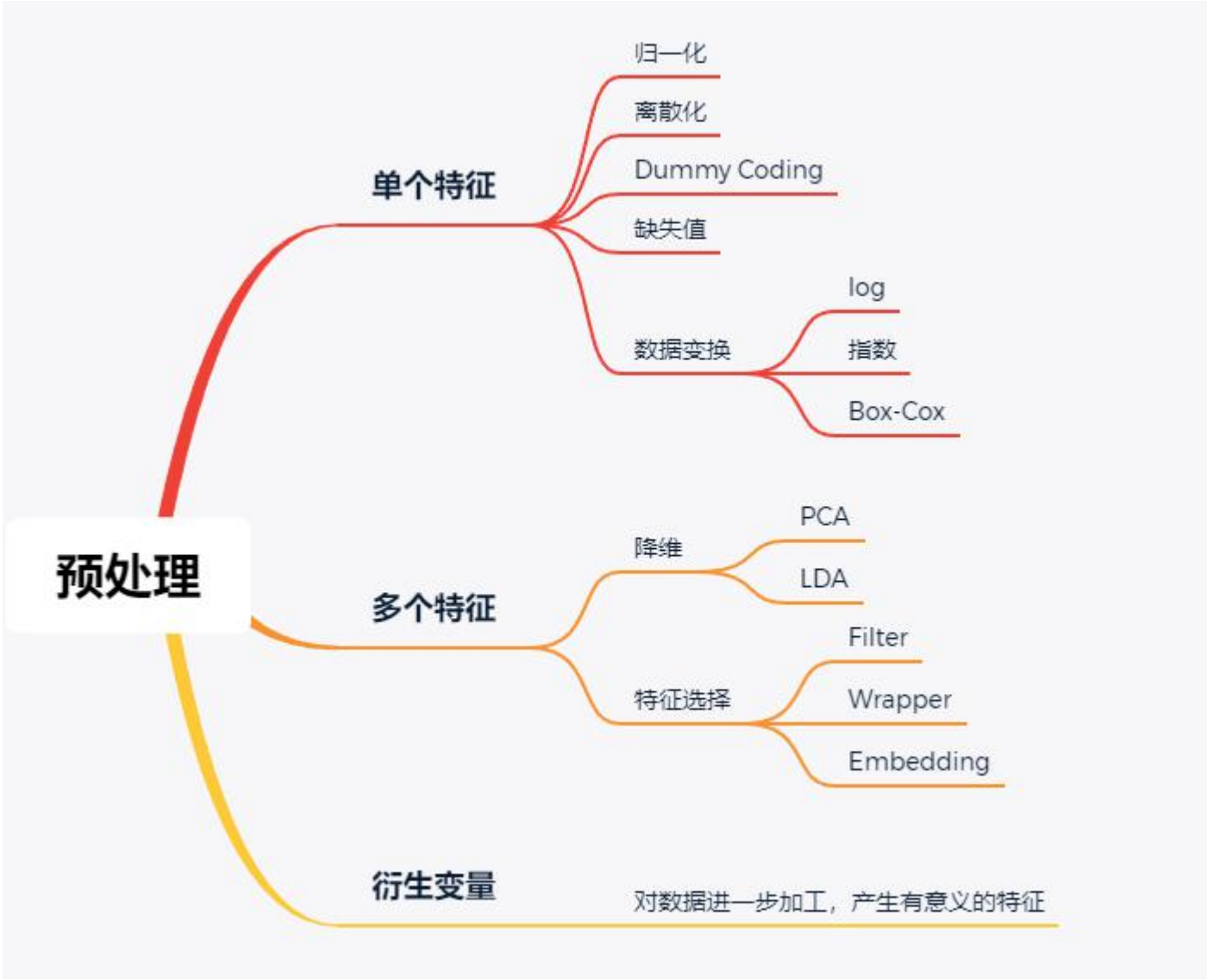


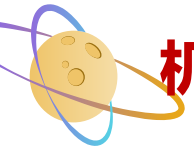


06

机器学习预处理

机器学习的预处理


预处理
单个特征
归一化
离散化
Dummy Coding
缺失值
数据变换
log
指数
Box-Cox
多个特征
降维
PCA
LDA
特征选择
Filter
Wrapper
Embedding
衍生变量
对数据进一步加工，产生有意义的特征

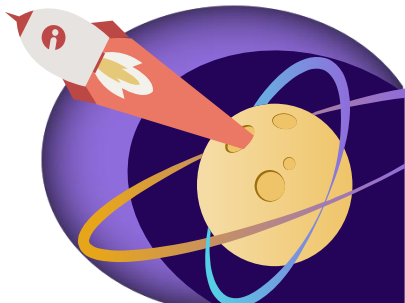


机器学习的预处理-离散化

LabelEncoder：

LabelEncoder是将labels转为数字

```
from sklearn import preprocessing
le = preprocessing.LabelEncoder()
le.fit([1, 2, 2, 6])
le.classes_
le.transform([1, 1, 2, 6])
le.inverse_transform([0, 0, 1, 2])
例2：
le = preprocessing.LabelEncoder()
le.fit(["paris", "paris", "tokyo", "amsterdam"])
list(le.classes_)
list(le.inverse_transform([2, 2, 1]))
```

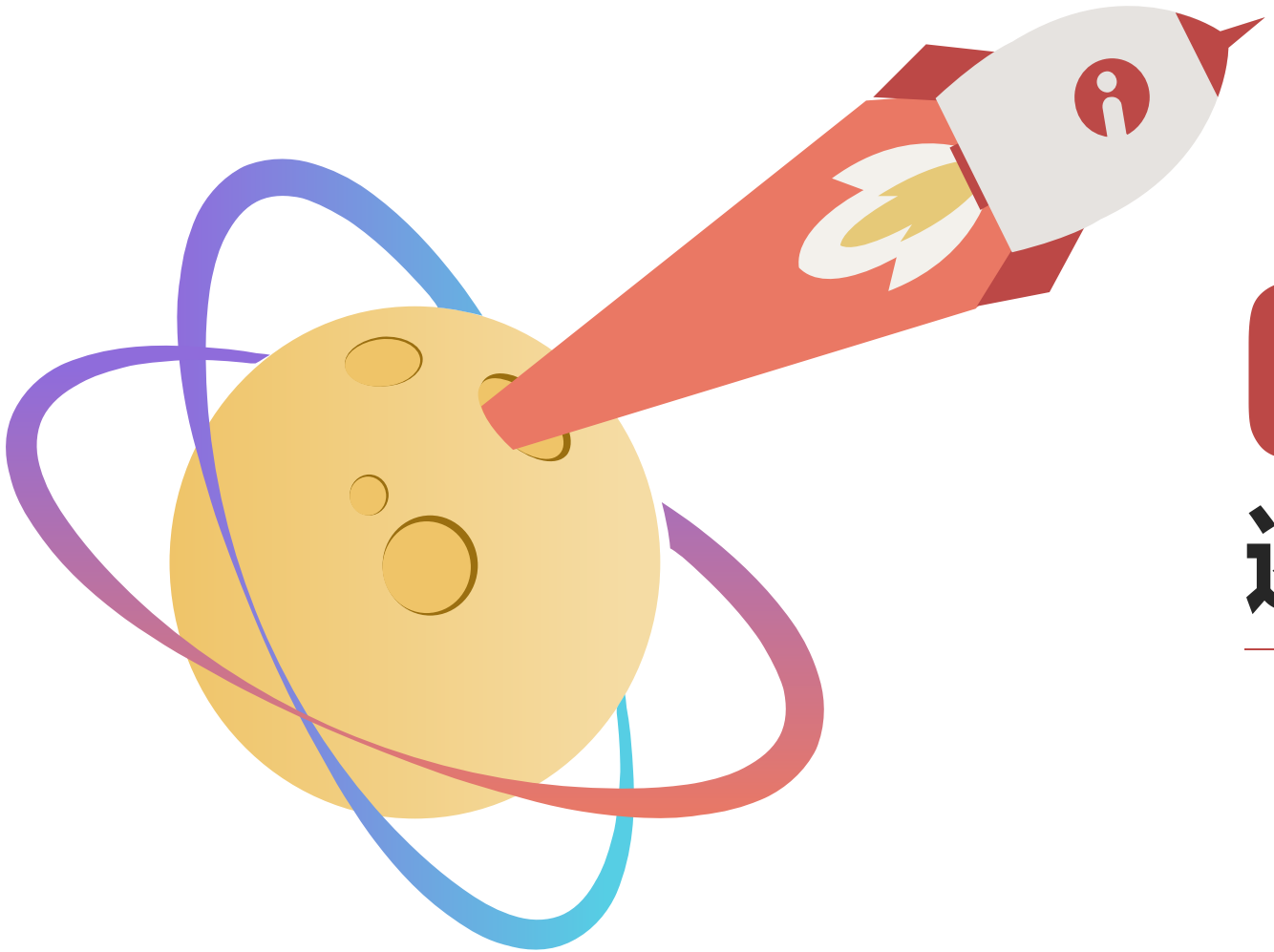


机器学习的预处理-DummyCodding

- One-Hot

假如有三种颜色特征：红、黄、蓝。 在利用机器学习的算法时一般需要进行向量化或者数字化。那么你可能想令 红=1，黄=2，蓝=3. 那么这样其实实现了标签编码，即给不同类别以标签。然而这意味着机器可能会学习到“红<黄<蓝”，但这并不是机器学习的本意，只是想让机器区分它们，并无大小比较之意。所以这时标签编码是不够的，需要进一步转换。因为有三种颜色状态，所以就有3个比特。即红色：１００，黄色: ０１０，蓝色：００１。

```
import pandas as pd
df = pd.DataFrame([
            ['green' , 'A'],
            ['red'   , 'B'],
            ['blue'  , 'A']])

df.columns = ['color',  'class']
pd.get_dummies(df)
```

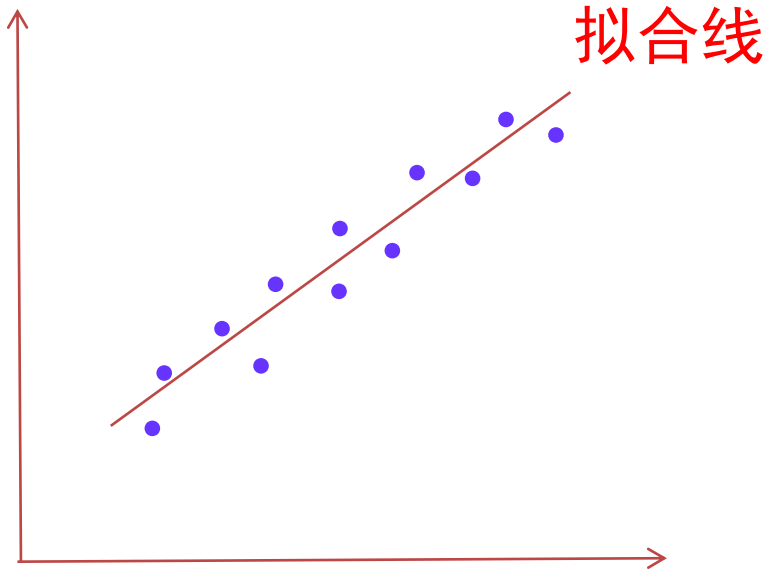


07

逻辑回归

线性回归原理

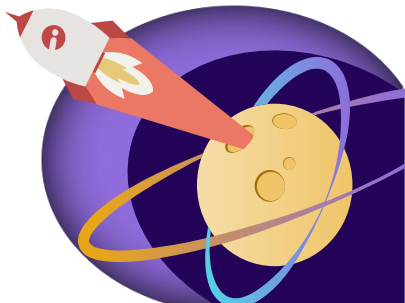
线性回归（Linear Regression）是一种通过属性的线性组合来进行预测的线性模型，其目的是找到一条直线或者一个平面或者更高维的超平面，使得预测值与真实值之间的误差最小化。



线性回归原理

$$h(x) = w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_nx_n$$

- 当只有一个 x_1时，$h(x)$为直线
- 当有x_1，x_2两个变量的时候，$h(x)$为一个平面
- 当有更多变量时，$h(x)$为高维的



线性回归原理

$$J = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - h(x_i)\right)^2$$

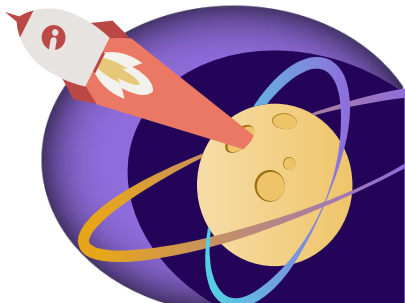
$h(x)$的预测值会和真实值会有所偏差，真实统计和$h(x)$预测数据的差称为残差。残差有正的有负的，为了降低计算复杂性，我们使用这个差值的平方进行计算。

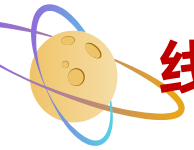
为了获得最好的 ,保证个点与实际数据的残差平方的总和最小。

1）偏导法

2）正规方程法

3）梯度下降 等等





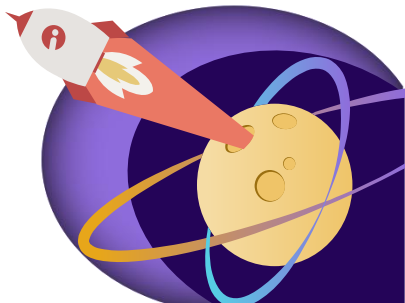
线性回归优缺点

优点

有很好的解释性，w权重可以看作是每个特征x的重要性

缺点

非线性数据拟合不好



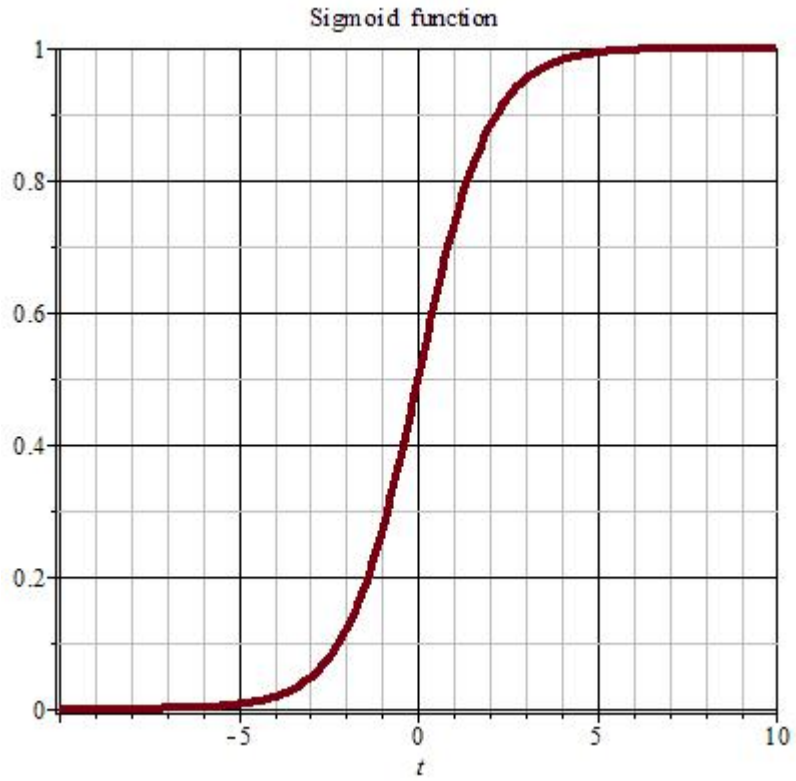
逻辑回归

连续值 → 0~1概率值 → 二分类

$z = h(x)$

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$\begin{cases} g(z) \geq 0.5 \\ g(z) < 0.5 \end{cases}$$



逻辑回归损失函数

$$J(\theta)=\frac{1}{2n}\sum_{i=1}^{n}\left(h_\theta(x^{(i)})-y^{(i)}\right)^2=\frac{1}{2n}\sum_{i=1}^{n}\left(\frac{1}{1+e^{-\theta^T x_i-b}}-y^{(i)}\right)^2$$

非凸，不好优化

标签为1的样本预测概率越接近1，损失越小；

标签为0的样本预测概率越接近0，损失越小。

最大似然函数 $$L=\prod_{i}^{N}h(x_i)^{y_i}*(1-h(x_i))^{1-y_i}$$

逻辑回归损失函数

$$L(w) = \sum_i (y_i * \log h(x_i) + (1 - y_i) \log(1 - h(x_i)))$$

$$L(w) = \sum y_i \left(\log h(x_i) - \log(1 - h(x_i))\right) + \log\left(1 - h(x_i)\right)$$

$$L(w) = \sum y_i \left(\log \frac{h(x_i)}{1 - h(x_i)}\right) + \log\left(1 - h(x_i)\right)$$

$$L(w) = \sum y_i \left(\log \frac{\frac{1}{1 + e^{-w^T X}}}{1 - \frac{1}{1 + e^{-w^T X}}}\right) + \log\left(1 - \frac{1}{1 + e^{-w^T X}}\right)$$

$$L(w) = \sum y_i \left(\log \frac{\frac{1}{1 + e^{-w^T X}}}{\frac{e^{-w^T X}}{1 + e^{-w^T X}}}\right) + \log\left(1 - \frac{1}{1 + e^{-w^T X}}\right)$$

$$L(w) = \sum y_i (w^T X) + \log\left(1 - \frac{1}{1 + e^{-w^T X}}\right)$$

$$L(w) = \sum y_i (w^T X) + \log\left(\frac{e^{-w^T X}}{1 + e^{-w^T X}}\right)$$

$$L(w) = \sum y_i (w^T X) + \log\left(\frac{1}{1 + e^{w^T X}}\right)$$

$$L(w) = \sum \left(y_i (w^T X) - \log(1 + e^{w^T X})\right)$$

权值优化

损失函数 $$J(w) = -\frac{1}{n}L(w)$$

两边对w求导

$$\frac{dJ}{dw} = yx - \frac{1}{1+e^{wx}} * e^{wx} * x$$

$$\frac{dJ}{dw} = x(y - h(x))$$

迭代权值优化 $$\theta_j = \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right) x_i^j$$

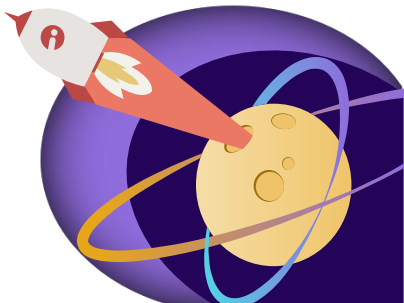
逻辑回归优缺点

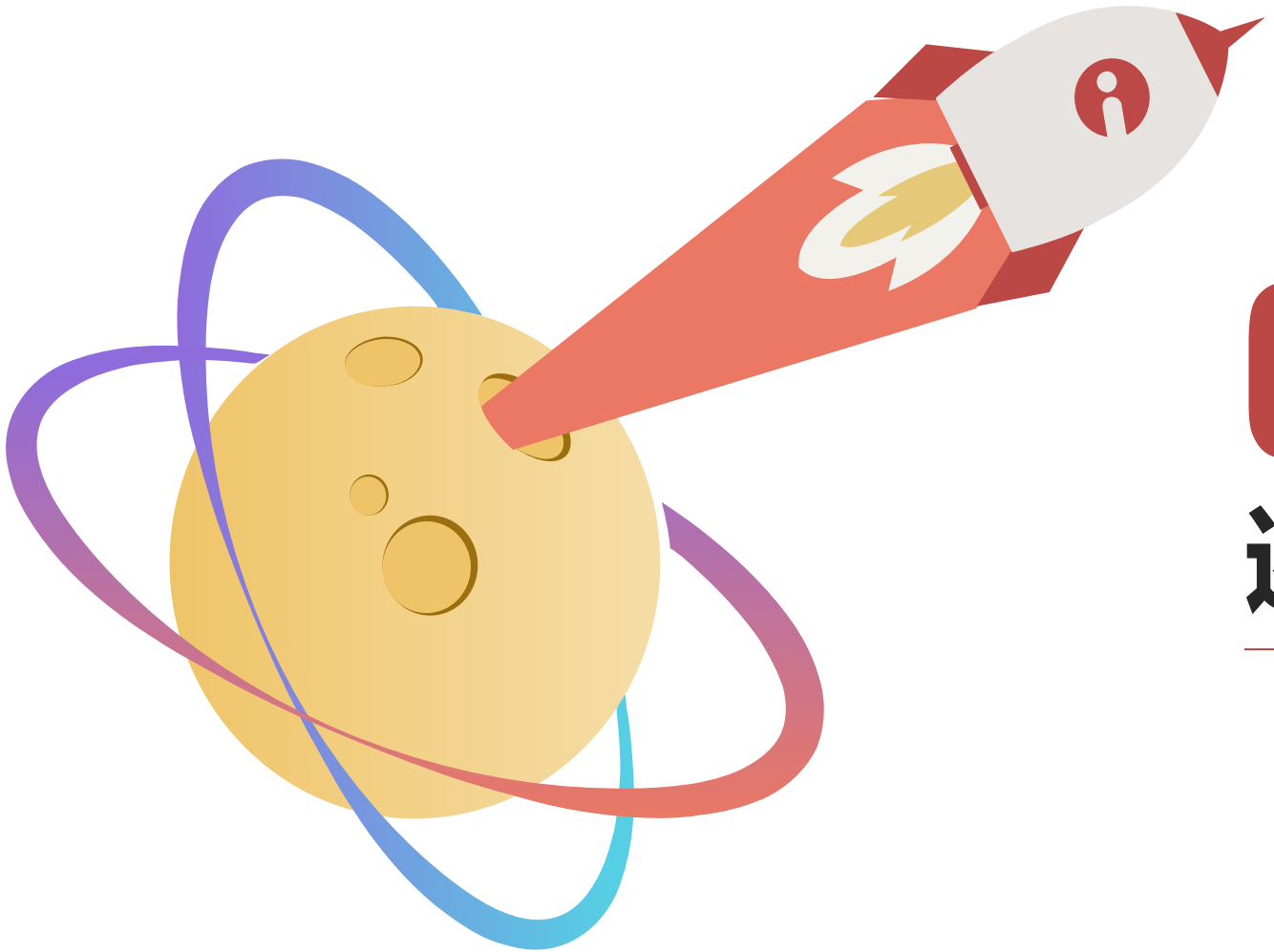
优点

- 容易理解和实现，可以观测样本的概率分数
- 训练速度快
- 由于经过了sigmoid函数的映射，对数据中小噪声的鲁棒性较好
- 不受多重共线性的影响(可通过正则化进行消除)

缺点

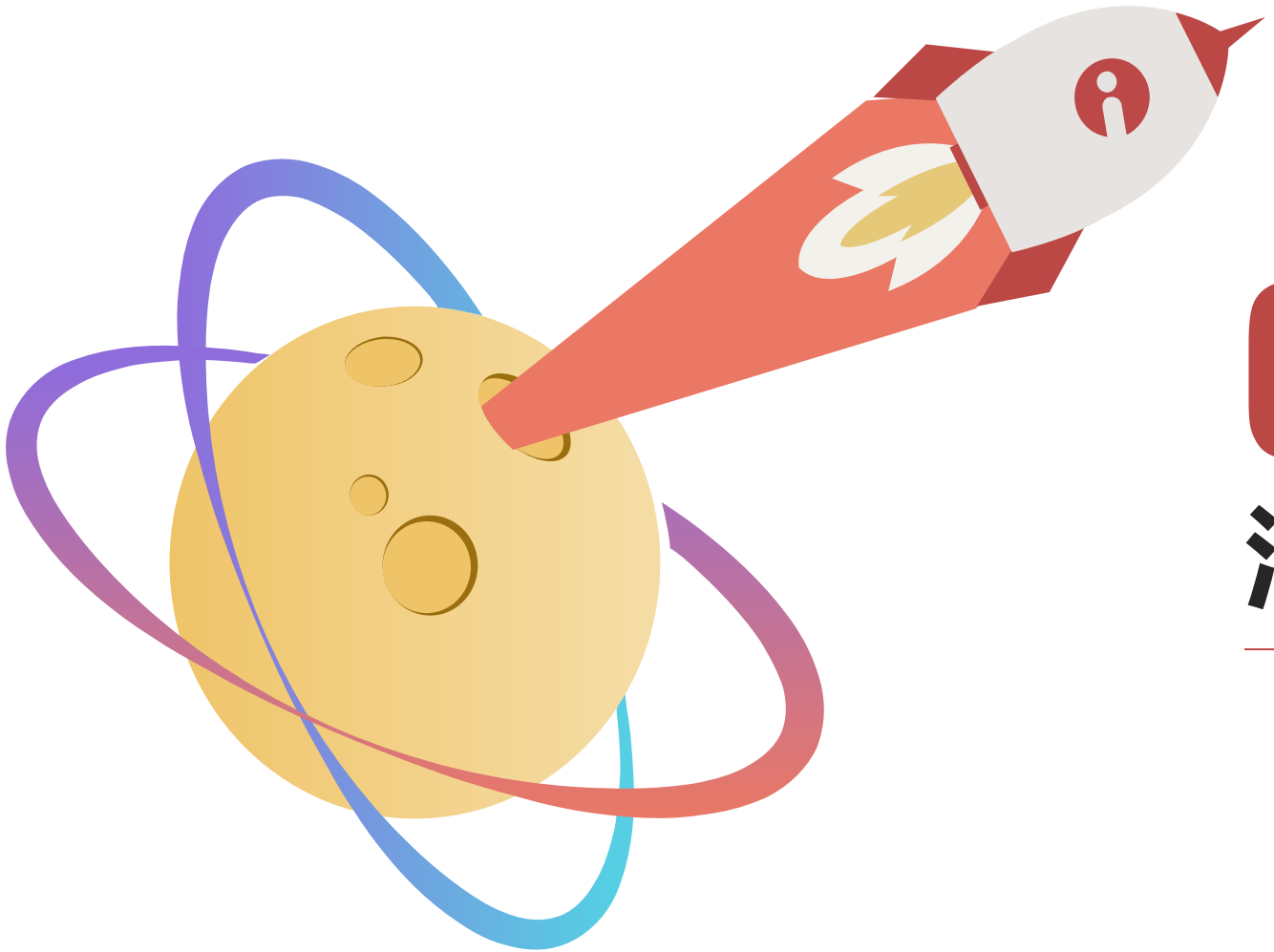
- 容易欠拟合
- 特征空间很大时效果不好
- 由于 函数s的特性，接近0/1的两侧概率变化较平缓，中间概率敏感，波动较大；导致很多区间特征变量的变化对目标概率的影响没有区分度，无法确定临界值





08

逻辑回归代码实现

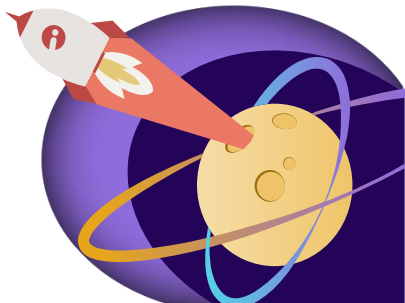


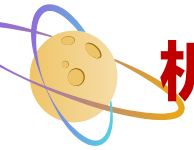
09

汇总

小结

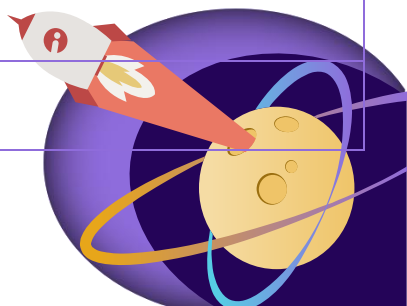
- Step1引用工具包
- Step2数据预处理：方便后续处理
- Step3选择模型，比如LR
- Step4训练模型（训练集）
- Step5模型评估（测试集）





机器学习算法工具包

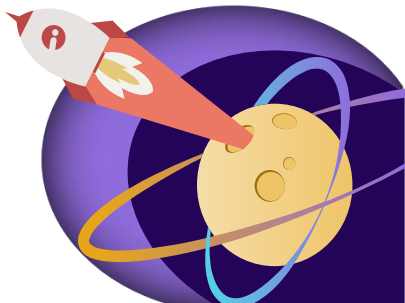
算法	工具
决策树	from sklearn.tree import DecisionTreeClassifier
朴素贝叶斯	from sklearn.naive_bayes import MultinomialNB
SVM	from sklearn.svm import SVC
KNN	from sklearn.neighbors import KNeighborsClassifier
Adaboost	from sklearn.ensemble import AdaBoostClassifier
K-Means	from sklearn.cluster import KMeans
EM	from sklearn.mixture import GMM
Apriori	from efficient_apriori import apriori
PageRank	import networkx as nx



机器学习的模型

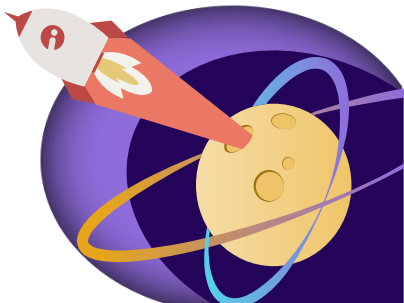
——10大经典模型

- 分类算法：C4.5，朴素贝叶斯（Naive Bayes），SVM，KNN，Adaboost，CART
- 聚类算法：K-Means，EM
- 关联分析：Apriori
- 连接分析：PageRank



作业1：使用CART算法对MNIST进行训练

- Step1 引用工具包
- Step2 数据预处理：方便后续处理
- Step3 选择模型，比如CART决策树
- Step4 训练模型（训练集）
- Step5 模型评估（测试集）



作业2：员工离职预测

IBM Watson是如何预测员工离职：

- https://www.dcjingsai.com/v2/cmptDetail.html?id=342
- 数据包括员工的各种统计信息，以及该员工是否已经离职，统计的信息包括了（工资、出差、工作环境满意度、工作投入度、是否加班、是否升职、工资提升比例等）
- 现在需要你来通过训练数据得出员工离职预测，并给出你在测试集上的预测结果

作业2：员工离职预测

数据表字段：

字段	定义
Age	员工年龄
Attrition	员工是否已经离职，1表示离职，2表示未离职
BusinessTravel	商务差旅频率，Non-Travel不出差，TravelRarely不经常出差，TravelFrequently经常出差
DailyRate	平均日工资
Department	员工所在部门，Sales销售部，Research & Development研发部，Human Resources人力资源部
DistanceFromHome	公司跟家庭住址的距离，从1到29，1表示最近，29表示最远
Education	员工的教育程度，从1到5，5表示教育程度最高
EducationField	员工所学习的专业领域，Life Sciences表示生命科学，Medical表示医疗，Marketing表示市场营销，Technical Degree表示技术学位，Human Resources表示人力资源，Other表示其他
EmployeeNumber	员工号码
EnvironmentSatisfaction	员工对于工作环境的满意程度，从1到4，1的满意程度最低，4的满意程度最高
Gender	员工性别，Male表示男性，Female表示女性

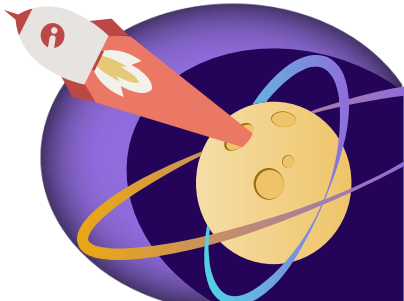
字段	定义
JobInvolvement	员工工作投入度，从1到4，1为投入度最低，4为投入度最高
JobLevel	职业级别，从1到5，1为最低级别，5为最高级别
JobRole	工作角色：Sales Executive销售主管，Research Scientist科学研究员，Laboratory Technician实验室技术员，Manufacturing Director制造总监，Healthcare Representative医疗代表，Manager经理，Sales Representative销售代表，Research Director研究总监，Human Resources人力资源
JobSatisfaction	工作满意度，从1到4，1代表满意度最低，4代表最高
MaritalStatus	员工婚姻状况，Single单身，Married已婚，Divorced离婚
MonthlyIncome	员工月收入，范围在1009到19999之间
NumCompaniesWorked	员工曾经工作过的公司数
Over18	年龄是否超过18岁
OverTime	是否加班，Yes表示加班，No表示不加班
PercentSalaryHike	工资提高的百分比

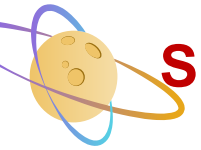
作业2：员工离职预测

数据表字段：

字段	定义
PerformanceRating	绩效评估
RelationshipSatisfaction	关系满意度，从1到4，1表示满意度最低，4表示满意度最高
StandardHours	标准工时
StockOptionLevel	股票期权水平
TotalWorkingYears	总工龄
TrainingTimesLastYear	上一年的培训时长，从0到6，0表示没有培训，6表示培训时间最长
WorkLifeBalance	工作与生活平衡程度，从1到4，1表示平衡程度最低，4表示平衡程度最高
YearsAtCompany	在目前公司工作年数
YearsInCurrentRole	在目前工作职责的工作年数
YearsSinceLastPromotion	距离上次升职时长
YearsWithCurrManager	跟目前的管理者共事年数

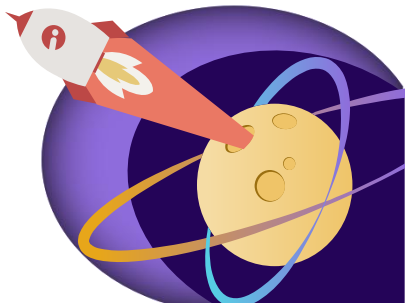
- 练习Logistic Regression

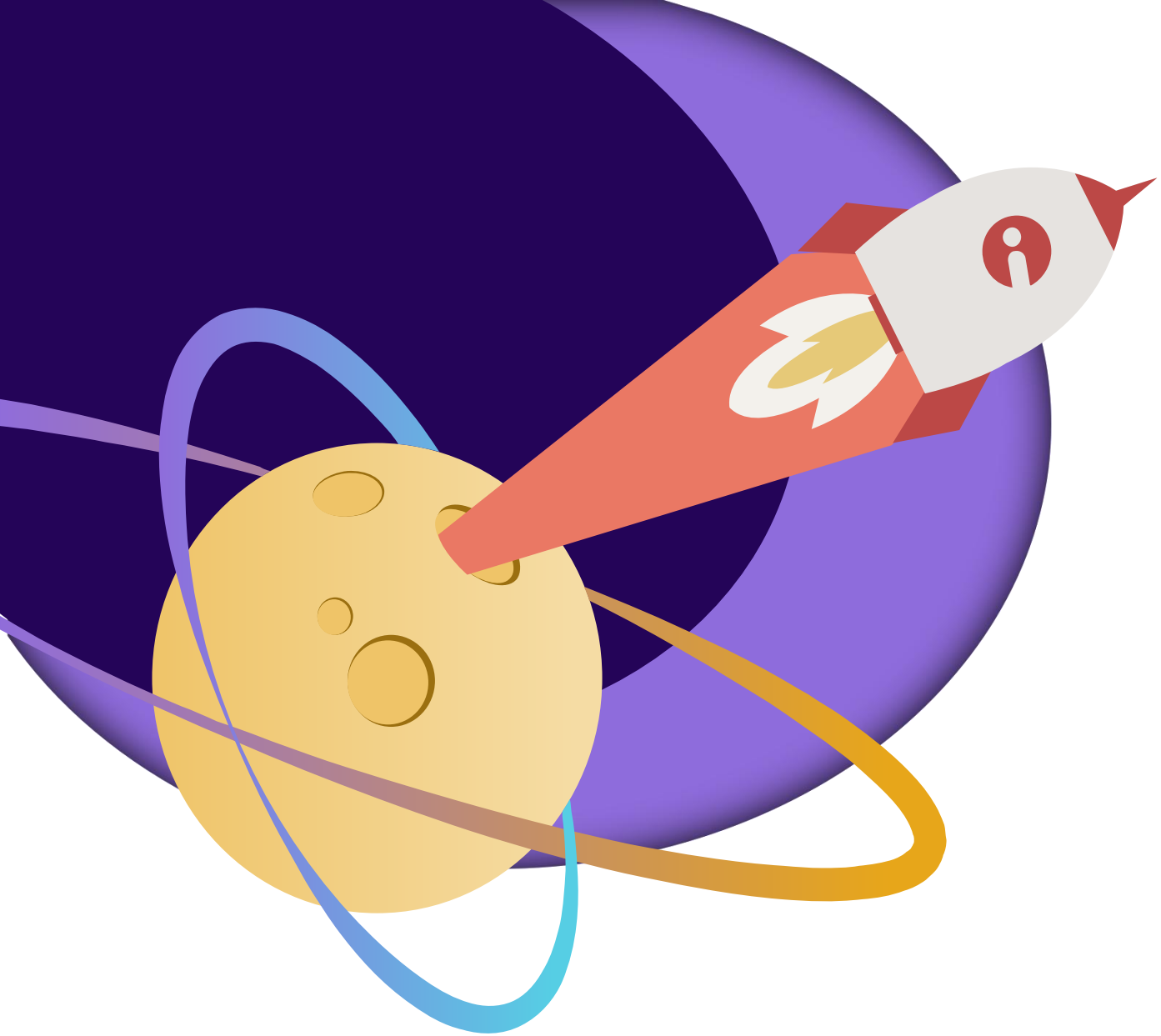




Summary

- 机器学习是关于预测的科学与技术
- 机器学习的7个步骤：收集数据，预处理，模型选择，训练，评估，超参数调整，预测
- 模型选择：传统机器学习 + 深度学习





Thank You