BUSINESS 2020

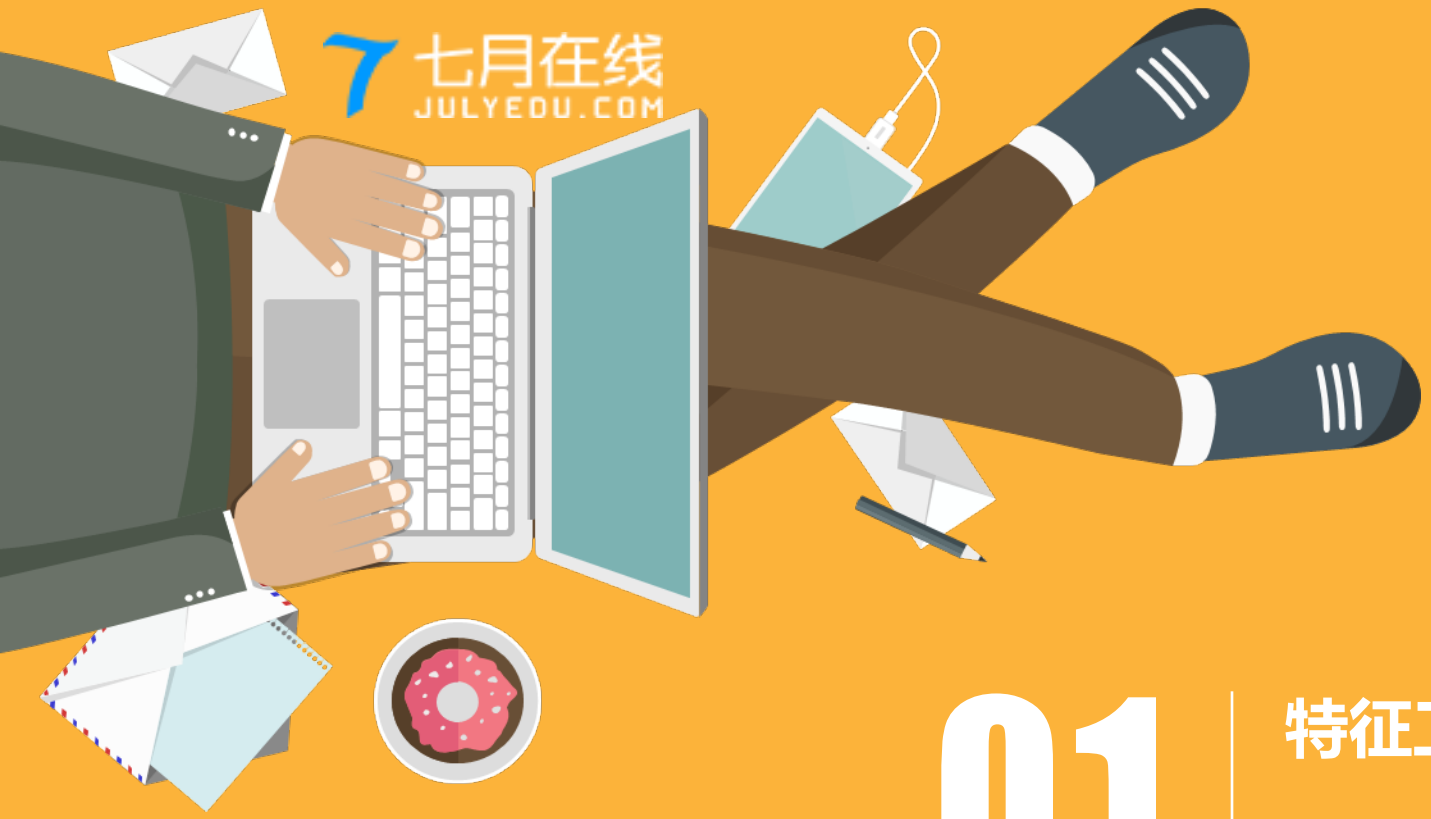
JULYEDU

从头到尾带打Kaggle比赛
特征工程

CV刘老师

https://www.julyedu.com/

CONTENTS



七月在线
JULYEDU.COM

01 特征工程介绍

Part 1 特征工程介绍

数据数量 比 **算法**重要，**数据质量** 比 **数据数量**更重要。

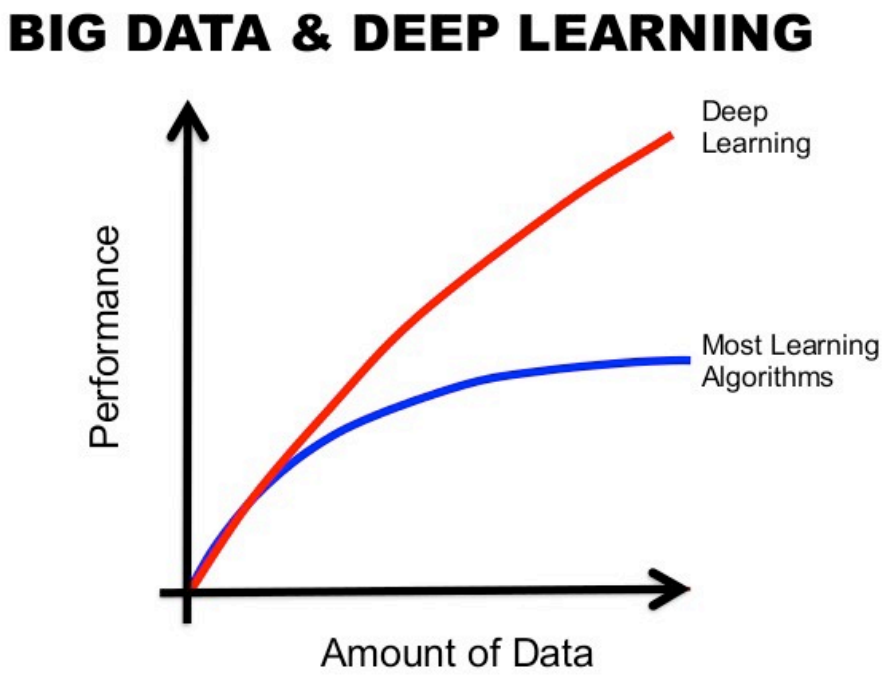
在大数据时代：

✓ 计算机的存储和计算不是问题，数据数量越来越多；

✓ 与传统机器学习相比，深度学习精度更高；

1、数据 & 特征工程决定了精度的上限；

2、机器学习模型只是不断逼近这个上限；



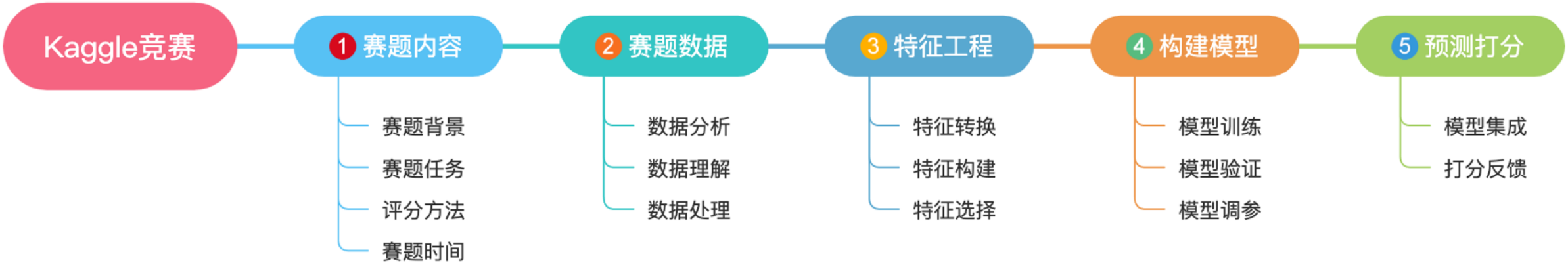
Part 1 特征工程介绍

特征工程是数据挖掘竞赛中的关键环节：

✓ 是一个需要有创造力的环节，需要有想象力的部分；

✓ 是一个需要思考、分析和验证的过程；

✓ 是一个需要耗费大量时间的过程；

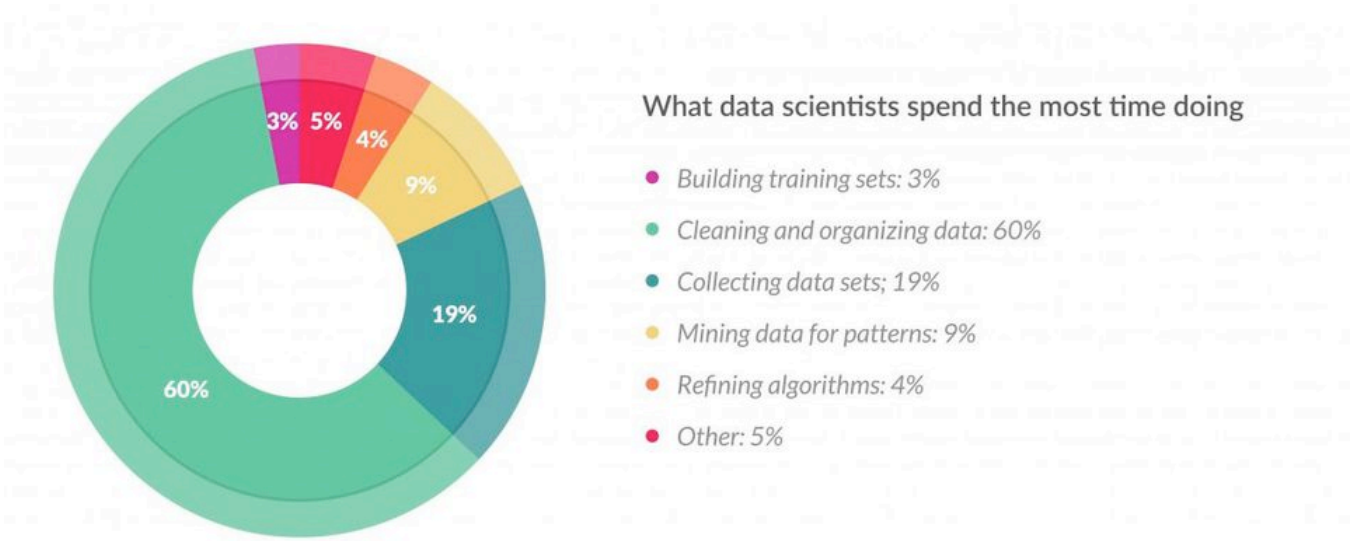


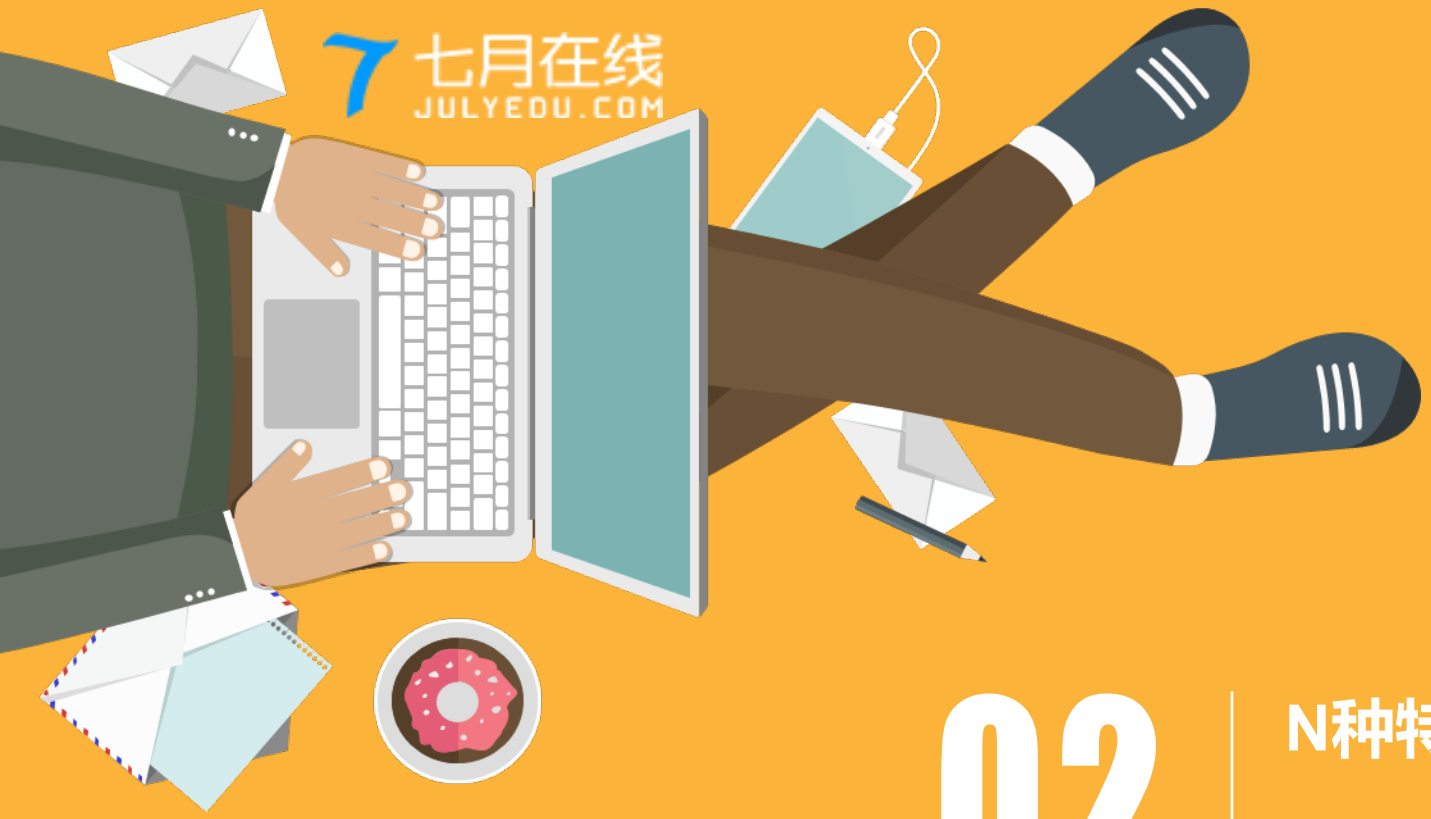
Part 1 特征工程介绍

特征工程步骤：

✓处理 & 清洗 数据；

✓转换 & 填充 数据；





七月在线
JULYEDU.COM

02 N种特征工程方法

Part 2 N种特征方法

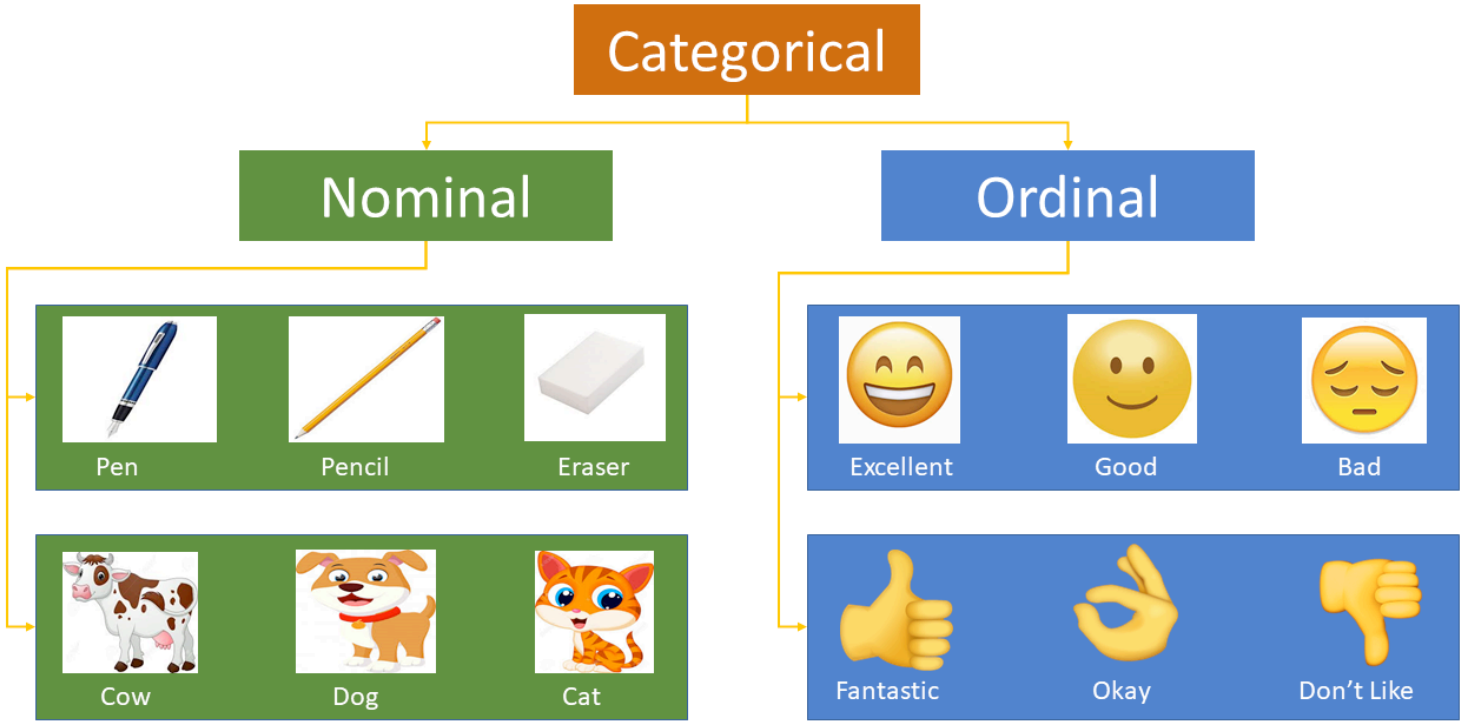
类别特征（Categorical Features）

- ✓ 是最常见的特征：
  - ✓ 个人信息：性别、城市、省份、名族、户口类型等；
  - ✓ 颜色：红色、白色、黑色、粉色等；
  - ✓ 国家：中国、美国、英国、新加坡等；
  - ✓ 动物：猫、狗、蛇、老虎、猴等；
- ✓ 任何时候都需要进行处理的数据；
- ✓ 高基数（High cardinality）会带来离散数据；
- ✓ 很难进行缺失值填充；

Part 2 N种特征方法

类别特征（Categorical Features）

✓ 可分类两种类型：无序（Nominal）和 有序（Ordinal）



Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

```
df = pd.DataFrame({
    'student_id': [1,2,3,4,5,6,7],
    'country': ['China', 'USA', 'UK', 'Japan', 'Korea', 'China', 'USA'],
    'education': ['Master', 'Bachelor', 'Bachelor', 'Master', 'PHD', 'PHD', 'Bachelor'],
    'target': [1, 0, 1, 0, 1, 0, 1]
})
df.head(10)
```

executed in 14ms, finished 13:19:51 2020-08-02

	student_id	country	education	target
0	1	China	Master	1
1	2	USA	Bachelor	0
2	3	UK	Bachelor	1
3	4	Japan	Master	0
4	5	Korea	PHD	1
5	6	China	PHD	0
6	7	USA	Bachelor	1

Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

- ✓ One Hot Encoding
- ✓ Label Encoding
- ✓ Ordinal Encoding
- ✓ Helmert Encoding
- ✓ Binary Encoding
- ✓ Frequency Encoding
- ✓ Mean Encoding
- ✓ Weight of Evidence Encoding
- ✓ Probability Ratio Encoding
- ✓ Hashing Encoding
- ✓ Backward Difference Encoding
- ✓ Leave One Out Encoding
- ✓ James-Stein Encoding
- ✓ M-estimator Encoding
- ✓ Thermometer Encoder

Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

✓ One Hot Encoding（独热编码）

- 形式：编码为One-of-K的K维向量形式；
- 用途：在所有的线性模型；
- 优点：简单，能够将类别特征进行有效编码；
- 缺点：会带来维度爆炸和特征稀疏；
- 实现方法：
  - 在pandas中使用get_dummies；
  - 在sklearn中使用OneHotEncoder；

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

executed in 14ms, finished 13:21:27 2020-08-02

	student_id	country	target	education_Bachelor	education_Master	education_PHD
0	1	China	1	0	1	0
1	2	USA	0	1	0	0
2	3	UK	1	1	0	0
3	4	Japan	0	0	1	0
4	5	Korea	1	0	0	1
5	6	China	0	0	0	1
6	7	USA	1	1	0	0

Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

✓ Label Encoding（标签编码）

- 形式：将每个类别变量使用独立的数字ID编码
- 用途：在树模型中比较适合；
- 优点：简单，不增加类别的维度；
- 缺点：会改变原始标签的次序关系；
- 实现方法：
  - pandas中的facotrize
  - sklearn中的LabelEncoder

```
df['country_LabelEncoder'] = pd.factorize(df['country'])[0]
df.head(10)
```

executed in 10ms, finished 13:34:47 2020-08-02

	student_id	country	education	target	country_LabelEncoder
0	1	China	Master	1	0
1	2	USA	Bachelor	0	1
2	3	UK	Bachelor	1	2
3	4	Japan	Master	0	3
4	5	Korea	PHD	1	4
5	6	China	PHD	0	0
6	7	USA	Bachelor	1	1

```
pd.factorize(df['country'])
```

executed in 7ms, finished 13:34:49 2020-08-02

```
(array([0, 1, 2, 3, 4, 0, 1]),
 Index(['China', 'USA', 'UK', 'Japan', 'Korea'], dtype='object'))
```

Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

✓ Ordinal Encoding（顺序编码）

- 形式：按照类别大小关系进行编码
- 用途：在大部分场景都适用；
- 优点：简单，不增加类别的维度；
- 缺点：需要人工知识，且对未出现的数值不友好；
- 实现方法：手动定义字典映射；

```
df['education'] = df['education'].map(
                    {'Bachelor': 1,
                    'Master': 2,
                    'PHD': 3})
df.head(10)
```

executed in 11ms, finished 13:46:25 2020-08-02

	student_id	country	education	target
0	1	China	2	1
1	2	USA	1	0
2	3	UK	1	1
3	4	Japan	2	0
4	5	Korea	3	1
5	6	China	3	0
6	7	USA	1	1

Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

✓ Binary Encoding（二进制编码）

- 形式：将类别进行编码然后进行二进制编码；
- 用途：与OneHot类似；
- 优点：简单，增加特征维度较低；
- 缺点：会带来维度爆炸和特征稀疏；
- 实现方法：使用BinaryEncoder；

```
import category_encoders as ce
encoder = ce.BinaryEncoder(cols= ['country'])

pd.concat([df, encoder.fit_transform(df['country']).iloc[:, 1:]], axis=1)
```

executed in 23ms, finished 14:02:34 2020-08-02

	student_id	country	education	target	country_1	country_2	country_3
0	1	China	Master	1	0	0	1
1	2	USA	Bachelor	0	0	1	0
2	3	UK	Bachelor	1	0	1	1
3	4	Japan	Master	0	1	0	0
4	5	Korea	PHD	1	1	0	1
5	6	China	PHD	0	0	0	1
6	7	USA	Bachelor	1	0	1	0

Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

✓ Frequency Encoding、Count Encoding

- 形式：将类别出现的次数或频率进行编码
- 用途：在大部分情况下都通用
- 优点：简单，可以统计类别次数；
- 缺点：容易受到类别分布带来的影响；
- 实现方法：使用次数统计；

```
df['country_count'] = df['country'].map(df['country'].value_counts()) / len(df)
df.head(10)
```

executed in 22ms, finished 14:36:42 2020-08-02

	student_id	country	education	target	country_count
0	1	China	Master	1	0.285714
1	2	USA	Bachelor	0	0.285714
2	3	UK	Bachelor	1	0.142857
3	4	Japan	Master	0	0.142857
4	5	Korea	PHD	1	0.142857
5	6	China	PHD	0	0.285714
6	7	USA	Bachelor	1	0.285714

```
df['country_count'] = df['country'].map(df['country'].value_counts())
df.head(10)
```

executed in 13ms, finished 14:36:43 2020-08-02

	student_id	country	education	target	country_count
0	1	China	Master	1	2
1	2	USA	Bachelor	0	2
2	3	UK	Bachelor	1	1
3	4	Japan	Master	0	1
4	5	Korea	PHD	1	1
5	6	China	PHD	0	2
6	7	USA	Bachelor	1	2

Part 2 N种特征方法

类别特征（Categorical Features）编码方式：

✓ Mean/Target Encoding

- 形式：将类别对应的标签概率进行编码；
- 用途：在大部分场景都可以通用；
- 优点：让模型更容易学习标签信息；
- 缺点：容易过拟合；
- 实现方法：使用次数统计；

```
df['country_target'] = df['country'].map(df.groupby(['country'])['target'].mean())
df.head(10)
```

executed in 12ms, finished 14:49:42 2020-08-02

	student_id	country	education	target	country_target
0	1	China	Master	1	0.5
1	2	USA	Bachelor	0	0.5
2	3	UK	Bachelor	1	1.0
3	4	Japan	Master	0	0.0
4	5	Korea	PHD	1	1.0
5	6	China	PHD	0	0.5
6	7	USA	Bachelor	1	0.5

Part 2 N种特征方法

数值特征（Numerical Features）

- ✓ 是常见的连续特征：
  - ✓ 年龄：18、19、25、40；
  - ✓ 成绩：55、60、75、80、95；
  - ✓ 经纬度：45.87、23.89、21.21；
- ✓ 容易出现异常值和离群点；

Part 2 N种特征方法

数值特征（Numerical Features）编码方式：

✓ Round

- 形式：将数值进行缩放、取整；
- 用途：在大部分场景都可以通用；
- 优点：可以保留数值大部分信息；
- 缺点：
- 实现方法：

```
df['age_round1'] = df['age'].round()
df['age_round2'] = (df['age'] / 10).astype(int)
df.head(10)
```

executed in 14ms, finished 15:20:54 2020-08-02

	student_id	country	education	age	target	age_round1	age_round2
0	1	China	Master	34.5	1	34.0	3
1	2	USA	Bachelor	28.9	0	29.0	2
2	3	UK	Bachelor	19.5	1	20.0	1
3	4	Japan	Master	23.6	0	24.0	2
4	5	Korea	PHD	19.8	1	20.0	1
5	6	China	PHD	29.8	0	30.0	2
6	7	USA	Bachelor	31.7	1	32.0	3

Part 2 N种特征方法

数值特征（Numerical Features）编码方式：

✓ Binning

- 形式：将数值进行分箱；
- 用途：在大部分场景都可以通用；
- 优点：可以将连续特征离散化
- 缺点：
- 实现方法：

```
df['age_<20'] = (df['age'] <= 20).astype(int)
df['age_20-25'] = ((df['age'] > 20) & (df['age'] <=25)).astype(int)
df['age_20-25'] = ((df['age'] > 25) & (df['age'] <= 30)).astype(int)
df['age_>30'] = (df['age'] > 30).astype(int)
df.head(10)
```

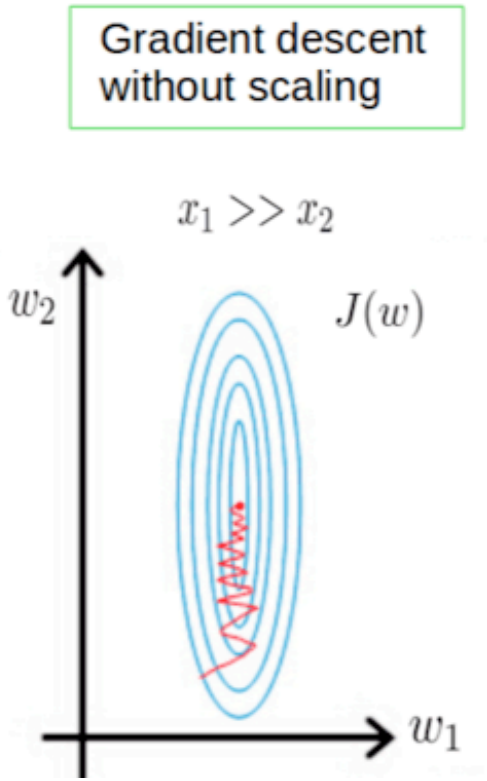
executed in 16ms, finished 15:26:29 2020-08-02

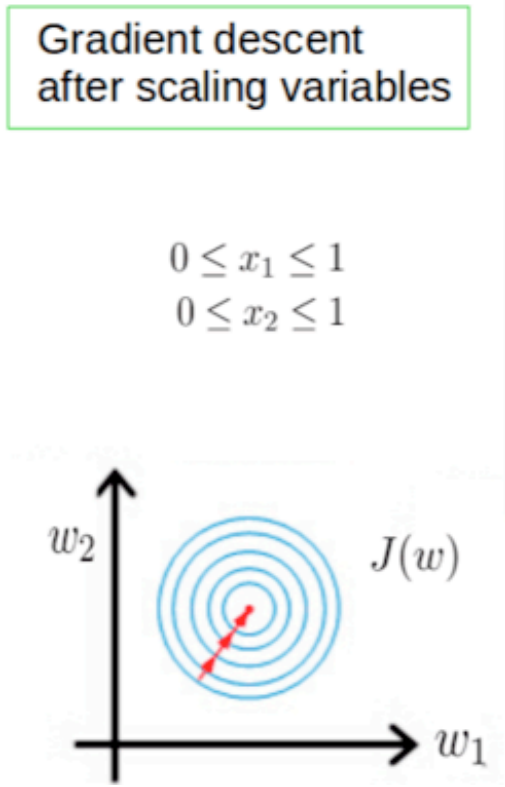
	student_id	country	education	age	target	age_<20	age_20-25	age_>30
0	1	China	Master	34.5	1	0	0	1
1	2	USA	Bachelor	28.9	0	0	1	0
2	3	UK	Bachelor	19.5	1	1	0	0
3	4	Japan	Master	23.6	0	0	0	0
4	5	Korea	PHD	19.8	1	1	0	0
5	6	China	PHD	29.8	0	0	1	0
6	7	USA	Bachelor	31.7	1	0	0	1

Part 2 N种特征方法

数值特征（Numerical Features）如何进行的特征缩放？

1) Min Max Scaler
2) Standard Scaler
3) Max Abs Scaler
4) Robust Scaler
5) Quantile Transformer Scaler
6) Power Transformer Scaler
7) Unit Vector Scaler





Part 2 N种特征方法

数值特征（Numerical Features）如何进行的特征缩放？

1）Min Max Scaler

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

- 形式：利用最大最小值进行处理；
- 用途：非高斯分布；
- 优点：可以保留数值大部分信息；
- 缺点：容易受到异常值影响；
- 实现方法：
  - from sklearn.preprocessing import MinMaxScaler

Part 2 N种特征方法

数值特征（Numerical Features）如何进行的特征缩放？

2）Standard Scaler

$$x_{new} = \frac{x - \mu}{\sigma}$$

- 形式：利用均值和方差进行处理；
- 用途：高斯分布；
- 优点：处理之后数据更加正态化；
- 缺点：对分布要求严格；
- 实现方法：
  - from sklearn.preprocessing import StandardScaler

Part 2 N种特征方法

数值特征（Numerical Features）如何进行的特征缩放？

3）Max Abs Scaler

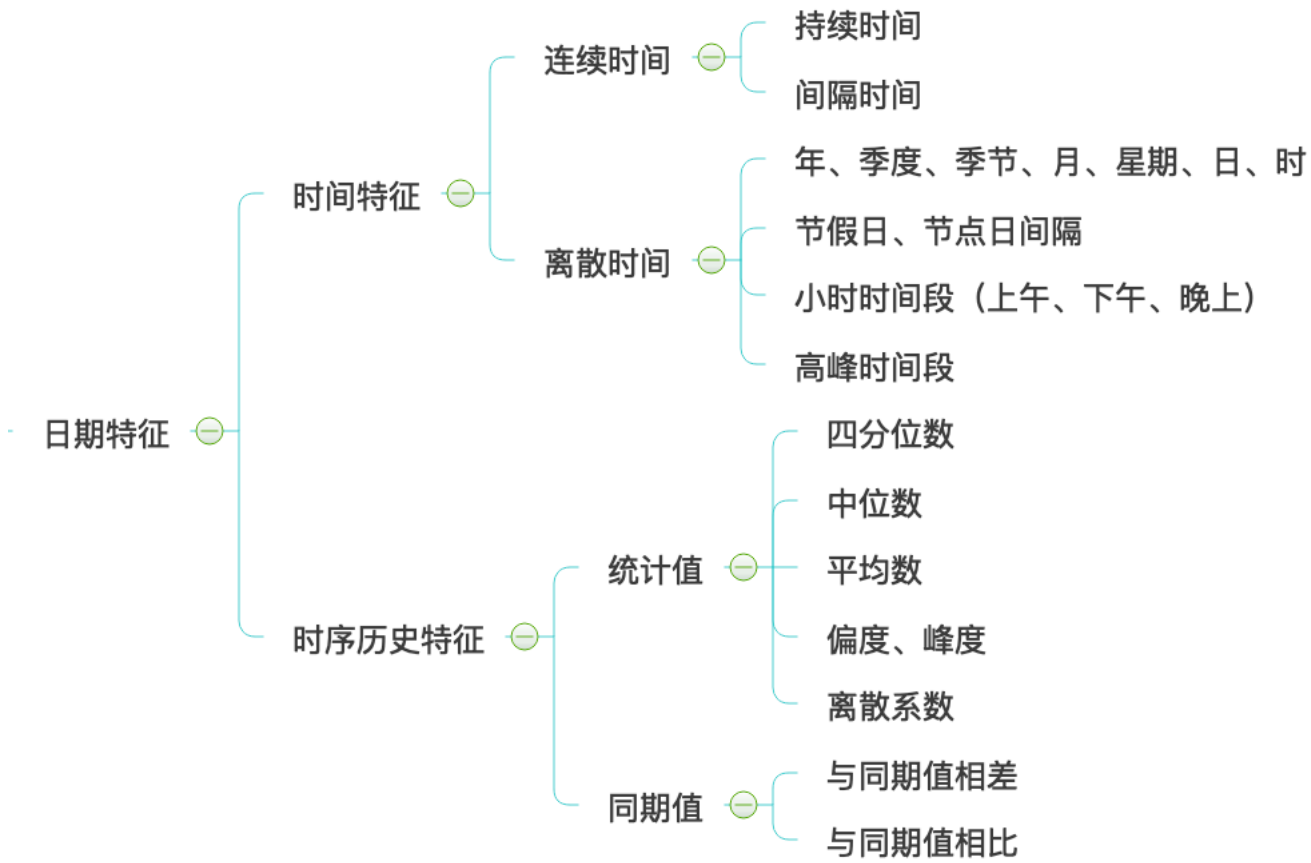
- 形式：与Min Max Scaler类似，但使用绝对最大只进行缩放；
- 用途：非高速分布
- 优点：
- 实现方法：
  - from sklearn.preprocessing import MaxAbsScaler

Part 2 N种特征方法

日期特征：

✓ 统计当前时间信息；

✓ 统计历史信息；



Part 2 N种特征方法

文本特征：典型的非结构数据

✓ 需要特殊对待，很容易得到稀疏数据；

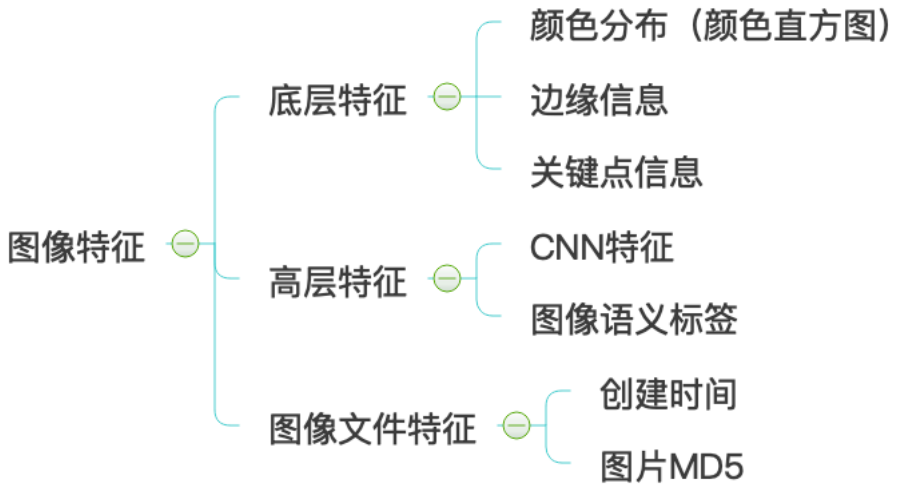
✓ 可以参考类别特征的处理方法；

Part 2 N种特征方法

图像特征：典型的非结构化数据

✓ 需要根据任务进行特定的提取；

✓ 可以参考图像处理的领域知识；



Part 2 N种特征方法

交叉特征：如何对特征进行交叉，构造新特征？

✓ 同类类型特征：加、减、除、笛卡尔积

✓ 不同类型特征：乘、除

✓ 聚合特征（先分组再聚合）：同比、环比

Part 2 N种特征方法

Leak & Golden特征：数据泄露，与赛题标签强相关的无效信息

例如：

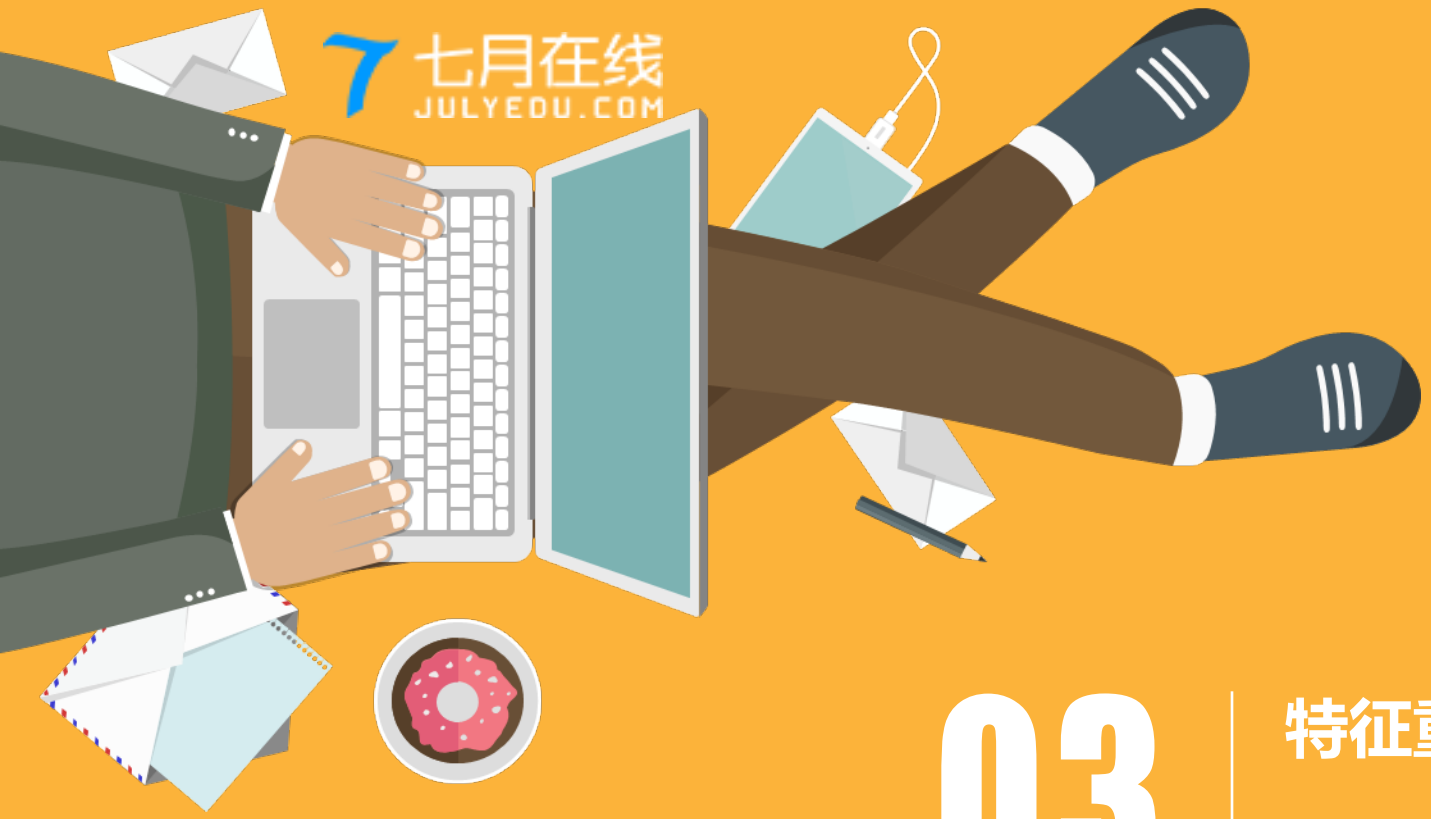
- ✓ 图像的创建时间、MD5信息；
- ✓ 不同类型样本的分布规律；

Part 2 N种特征方法

思考1：你真的掌握了上述的特征工程方法了吗？

思考2：不同的机器学习模型能学习到不同特征，是人工先做特征，还是让机器去学习？

思考3：如果是匿名特征，如何做特征工程？



七月在线
JULYEDU.COM

03 特征重要性 & 特征筛选

Part 3 特征重要性计算

随机森林：使用节点分裂的Gini指数来进行衡量；

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

```
rf = RandomForestRegressor()
rf.fit(data.data, data.target);
print(rf.feature_importances_)
```

Part 3 特征重要性计算

LightGBM： If “split”, result contains numbers of times the feature is used in a model. If “gain”, result contains total gains of splits which use the feature.

https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.plot_importance.html

```
rf = LGBMRegressor()
rf.fit(data.data, data.target);
print(rf.feature_importances_)
```

Part 3 特征重要性计算

XGboost： “gain”, “weight”, “cover”, “total_gain” or “total_cover”.

https://xgboost.readthedocs.io/en/latest//python/python_api.html?highlight=feature_importances_

```
rf = XGBRegressor()
rf.fit(data.data, data.target);
print(rf.feature_importances_)
```

Part 3 特征重要性计算

思考：模型输出的重要性 和 特征真实的重要性是一致的么？

Part 4 特征筛选方法

✓ Filter(过滤法)：在训练代码之前进行特征筛选，筛选步骤与后续训练模型无关；

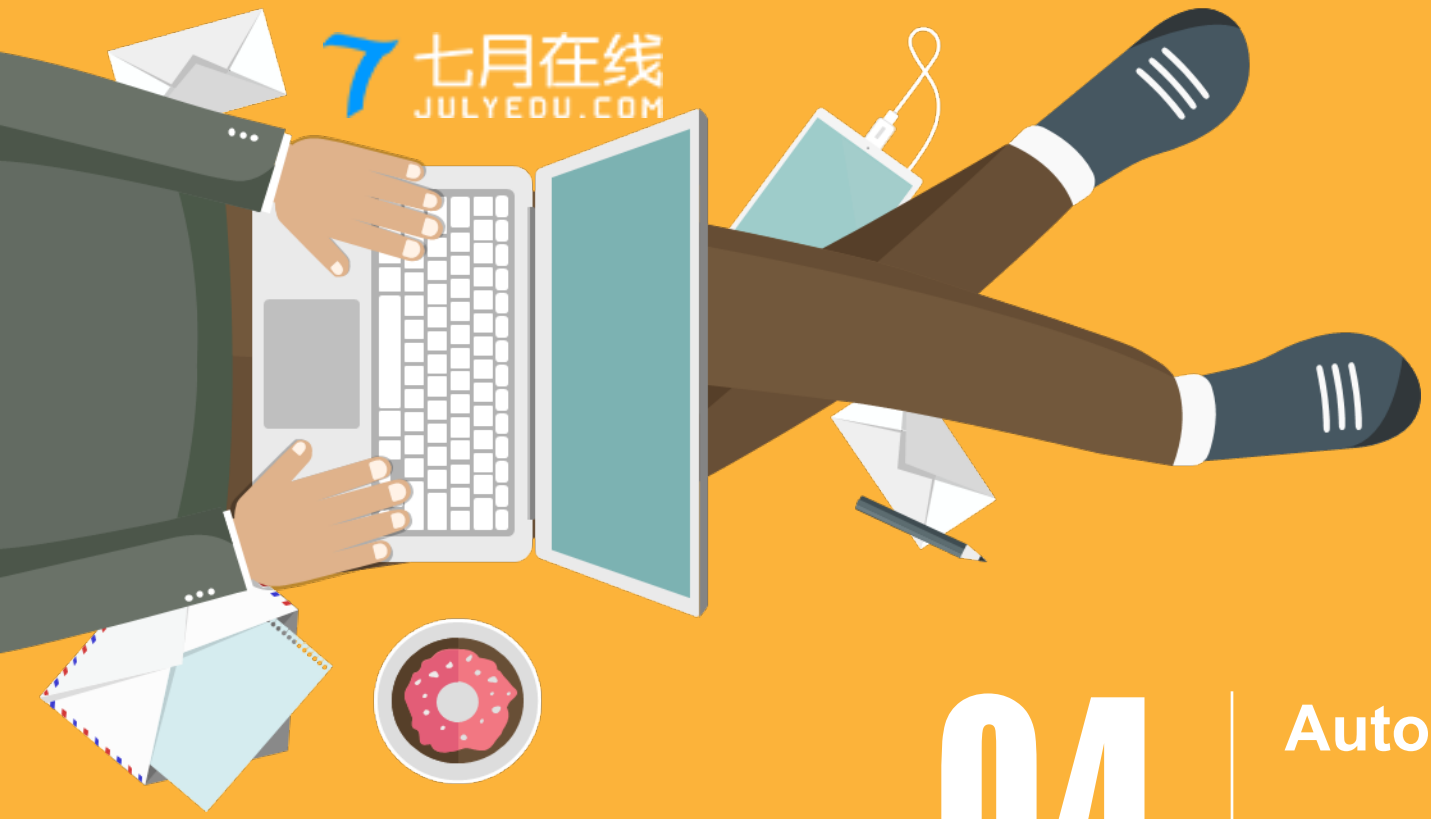
可以设定统计阈值 或 将待选择特征的个数进行筛选

✓ Wrapper(包装法)：通过训练与验证来找到合适的特征子集；

选择在验证集上精度最高的特征子集，在竞赛过程比较合适用；

✓ Embedded(嵌入法)：在训练过程中，利用模型参数 或 信息增益选择；

选择随机森林特征重要性前20个特征；

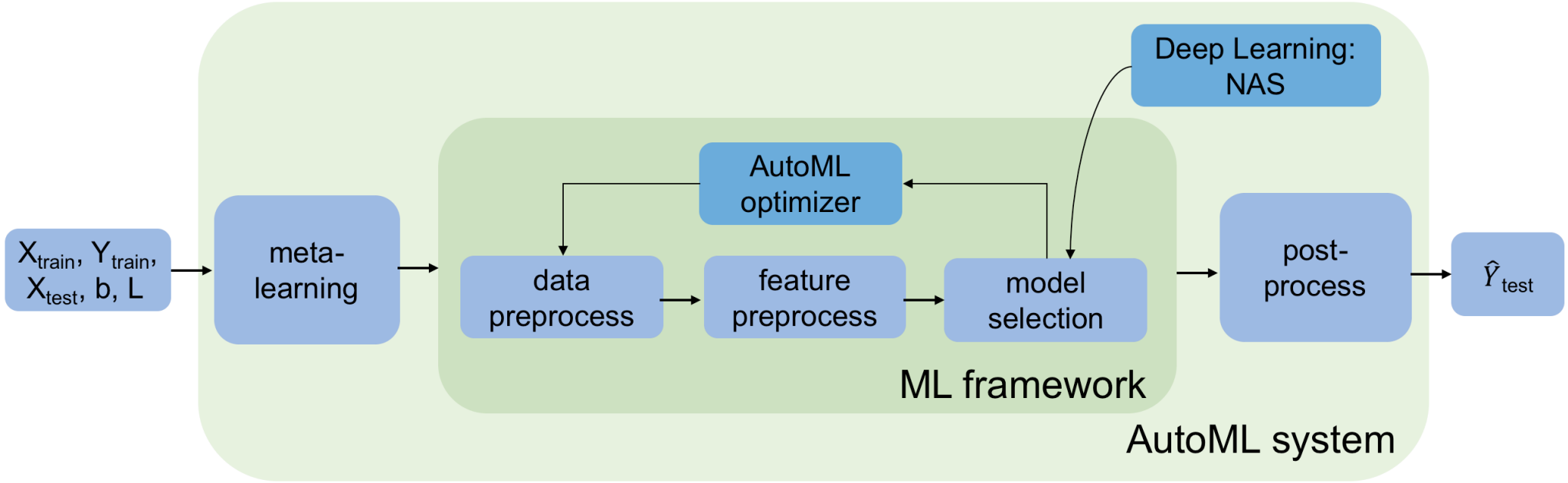


七月在线
JULYEDU.COM

04 AutoML

Part 4 AutoML

AutoML可以自己完成：特征工程、模型训练、模型调参的过程；



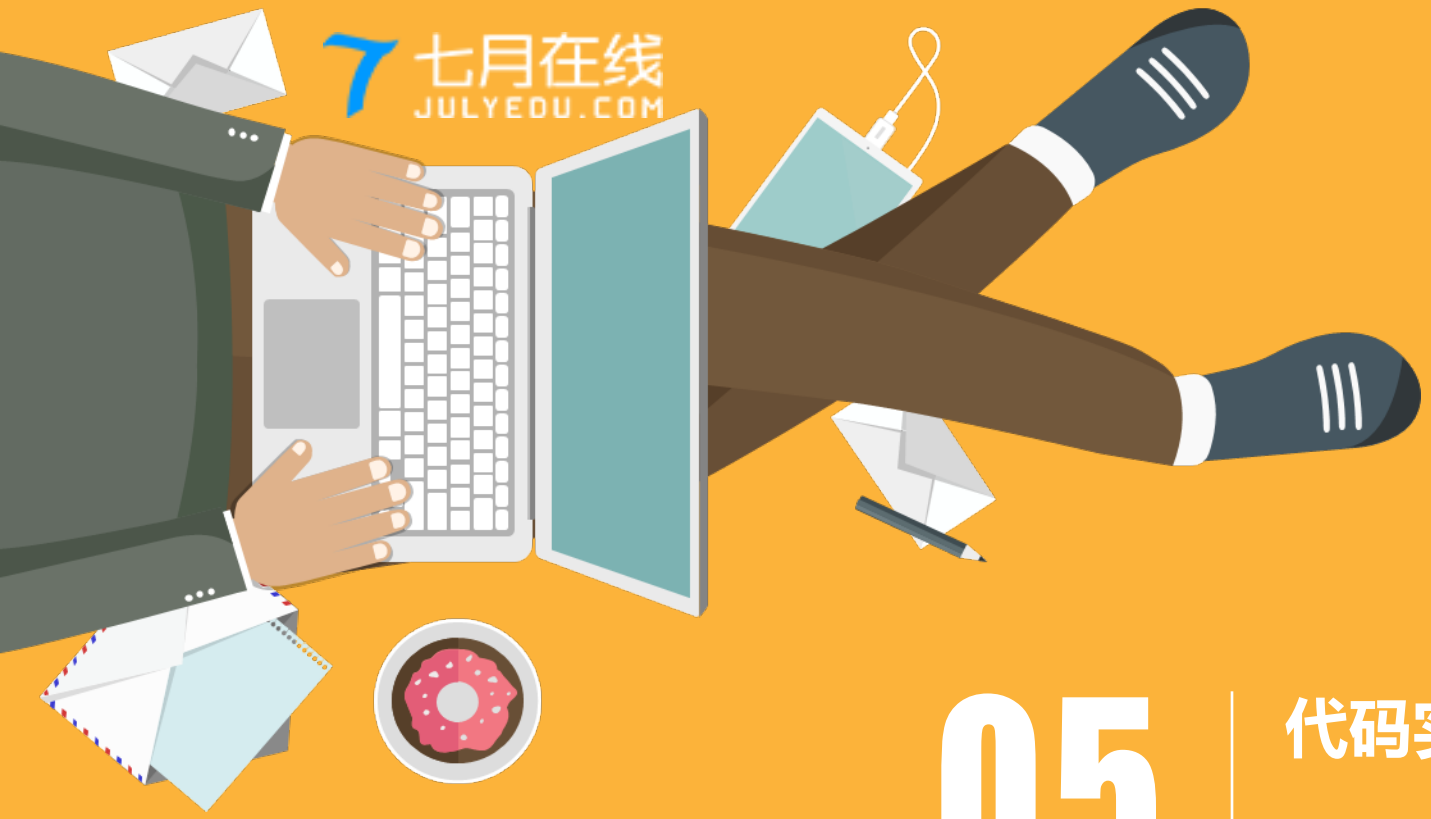
Part 4 AutoML

AutoML可以自己完成：特征工程、模型训练、模型调参的过程；

- Data preparation and ingestion (from raw data and miscellaneous formats)
  - Column type detection;
  - Column intent detection;
- Feature engineering
  - Feature selection
  - Feature extraction
  - Meta learning and transfer learning
  - Detection and handling of skewed data and/or missing values
- Model selection
- Hyperparameter optimization of the learning algorithm and featurization
- Selection of evaluation metrics and validation procedures
- Analysis of results obtained

Part 4 AutoML

思考：AutoML能代替算法工程么，算法工程师会失业么？



七月在线
JULYEDU.COM

05 代码实践

Part 5 代码实践

1、非结构化特征工程

https://www.kaggle.com/kashnitsky/topic-6-feature-engineering-and-feature-selection

2、leak特征

https://www.kaggle.com/c/two-sigma-connect-rental-listing-inquiries/discussion/31870

阅读链接

1、https://towardsdatascience.com/all-about-categorical-variable-encoding-305f3361fd02

2、https://towardsdatascience.com/all-about-missing-data-handling-b94b8b5d2184

3、https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35

4、https://towardsdatascience.com/understanding-feature-engineering-part-1-continuous-numeric-data-da4e47099a7b

课后作业

作业1、使用课程讲解的内容改进baseline思路，加入自己的特征；

作业2、加入基于manager_id的target encoding + Leak特征，并进行提交，截图发在群里；

微信扫一扫关注我们

THANKS

刘老师

https://www.julyedu.com/