*以 score 检验模型好坏

聚类算法（第三章）

Supervised Learning

多项式（第四章）

*如果不考虑内存与时间，适用大数据集

KNeighbors Classifier

KNeighbor 分类器
- kNN：根据测试点周围多个邻居的分类，投票决定测试点
- K 近邻回归：
- 优：易理解，速度快，缺：数据集大时处理速度慢，对多特征数据集效果不好

（回归）线性模型
- 斜率：coef　截距：intercept
- 线性回归：高维数据集可能过拟合
- 岭回归：令 w 所有元素均接近0。可以改变 alpha 改变正则化强度（L2正则化）
- Lasso：令某些元素 w 为0，被忽略 可能欠拟合（L1正则化）

Summerize：一般首选岭回归，若只有几个重要特征/需要易解释模型 -Lasso

（分类）线性模型
- Logistic
- 线性支持向量机 } 默认使用 L2正则化；改为 L1：penalty="l1"
  - 参数 C↑：正则化↓；C 小时适应大部分数据点；C 大时每个数据点分类正确
  - 高维数据
  - Summerize：对于大数据集训练速度快，预测速度快但很难对系数作出解释
  - 适用：大数据集；特征数量大于样本数量

朴素贝叶斯分类器：通过单独查看每个参数学习并统计数据
- 训练速度快 泛化能力差
- 适用高维稀疏数据
- Gaussian NB 任意连续数据
- Bernoulli NB 二分类数据
- Multinomial NB 计数数据

决策树：每个结点包含一个测试，递归划分
- 优：数据易于可视化；算法不受数据缩放影响
- 劣：经常过拟合、泛化能力差
- 限制 max_deep，减少问题以防过拟合
- graphviz 分析
- 或使用特征重要性分析

集成学习
- 随机森林：对所有树取平均值以降低过拟合
  - 优：数据易于可视化；算法不受数据缩放影响；不需调参
  - 劣：时间慢，内存大
  - ①确定树个数 ②自助采样
  - 不适用高维稀疏数据，建议用线性模型
- 递度提升回归树：每棵树试图纠正前一棵树的错
  - Learning rate：纠正强度
  - n_estimators：树数量→模型复杂度
  - 需要调参，训练时间长 不适用高维稀疏数据

核支持向量机
- 分类 svm
  - ①添加非线性特征（线性特征的交互项）
  - ②核技巧：计算扩展特征中数据点的距离
  - ③调参 gramma gramma 小，高斯核半径大，过拟合
  - 学习每个数据点对决策边界重要性
  - 得到支持向量
  - 在各种数据集上表现好，对样本个数缩放不好
  - 适用低、高维数据集
- 回归 SVR
  - 若特征数量级不同→缩放特征在 0~1
  - 预处理，调参难 检查不易

神经网络
- 多特征 加权 隐单元 加权 … 输出
- 适用低、高维数据集
- 优：模型复杂，数据量大
- 缺：训练时间长，数据预处理

# 单变量替换 (Chapter 4)

- Unsupervised Learning
  - 数据变换
    - PCA主成分分析
      - 过程
        - 找到方差最大的方向，标记为成分1
        - 找到与成分1 正交且包含信息最多的成分2
      - 优点
        - 去除噪声影响
        - 保留的部分信息可视化
        - 降维
    - NMF非负矩阵分解
      - 特征
        - 每个数据点写成一些分量的加权求和
        - 所有分量和系数均为非负
      - 优势
        - 解除抵消效应
      - 注意点
        - n_components<features，**否则每个特征可以直接解释**
    - t_SNE流形学习
      - 用途
        - 散点图可视化
      - 原理
        - 找到数据的一个二位表示，尽可能保持数据间距离
        - 距离近的点更靠近
        - 距离远的点更远离

用 fit_transform

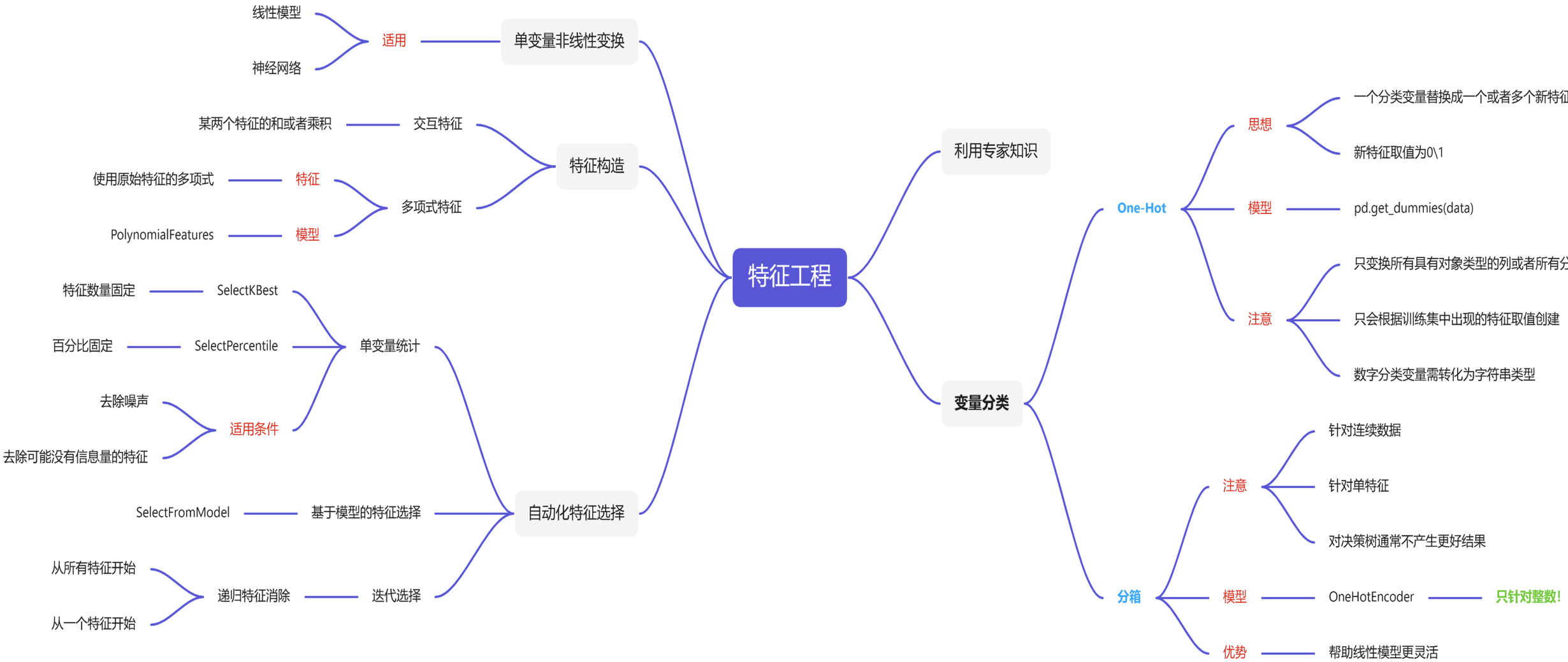
使用 transform

  - 预处理与缩放
    - StandardScaler
      - 特点：确保每个特征平均值为0，方差为1
      - 优势：所有特征在同一量级
      - 劣势：不能保证任何特定的最大值或者最小值
    - RobustScaler
      - 特点：使用中位数和四分位数缩放
      - 优势：可以忽略有较大误差的测量值，特征在同一量级
      - 劣势：不能保证任何特定的最大值或者最小值
    - MinMaxScaler
      - 特点：使得所有特征刚好位于0~1之间
    - Normalizer
      - 欧式长度：将数据点投射到半径为1的圆上
      - 特点：对每个数据点进行缩放，使得特征向量的欧式长度等于1
      - 适用：数据方向/角度重要，特征向量的长度无关紧要

Attention:
缩放器对 train 和 test
作同样变化

  - 聚类算法
    - k均值聚类
      - 过程
        - 找到簇中心
        - 分配数据点
        - 簇中心设置为新数据点的平均值
        - 分配不再变化则算法结束
      - 隐患
        - 假设簇有相同“直径”
        - 不能总是找到，即使簇的个数正确
        - 假设所有方向对簇同等重要
        - 无法识别有复杂形状的簇
        - 依赖随机初始化
      - 优势
        - 相对容易理解和实现
        - 运行速度快
        - 可以轻松扩展到大型数据集
    - 凝聚聚类
      - 声明每个点都是簇
      - 过程
        - 合并两个最相似的簇
      - 优势
        - 知道簇的个数满足要求
    - DBSCAN
      - 原理
        - 识别“拥挤”区域的点
        - 分隔密集区域
      - 优势
        - 能找出不属于任何簇的点
        - 可以划分具有复杂形状的簇
        - 不用设置簇的个数
      - 组成
        - 核心点
        - 边界点
        - 噪声
    - 对比与评估
      - 真实值评估聚类
        - ARI
        - NMI
      - 无真实值
        - 轮廓系数

评价一般
较为主观




- 分箱、多项式，交互项对复杂度低模型大有提升 线性模型，朴素贝叶斯模型
- mask: get_support 可视化选中数据

# 文本数据处理

bag-of-words

导入词袋 → 拟合 → CountVectorizer（分词 & 构建词表）

- Vocabulary_访问词表
- transform 训练词袋表示
- toarray 查看出现次数

CountVectorizer → 训练分类器（线性模型如 LogisticRegression）

只计算每个单词出现频次

正则表达式提取词例 \b\w\w+\b

所有字母转换为小写 优：无一个字母的单词

缺：有很多无信息的特征

优化

- min_df：至少在几个文档中出现
- stopword：列表
- 按预计信息量大小缩放特征 → 常用 → tf-idf 词频—逆同文档频率

tf-idf：在某个特定文档中经常出现则权重高；在多个文档出现则权重低

可使用网格搜索提高准确率

ngram_range=(x, y)

以 x 元分词到 y 元分词 → n 元分词

将 replaces, replacement, replaced, replacing 归为一类

tokenization

optimize 分词优化

- 词干提取 porter
- 词形还原 spacy

主题建模 → 分解 → Latent Divichlet Allocation → 查看主题

- print_topics
- document_topics：求和看到每个主题整体权重