基于大数据的机器学习实践
实 践 报 告

机器学习关键阶段

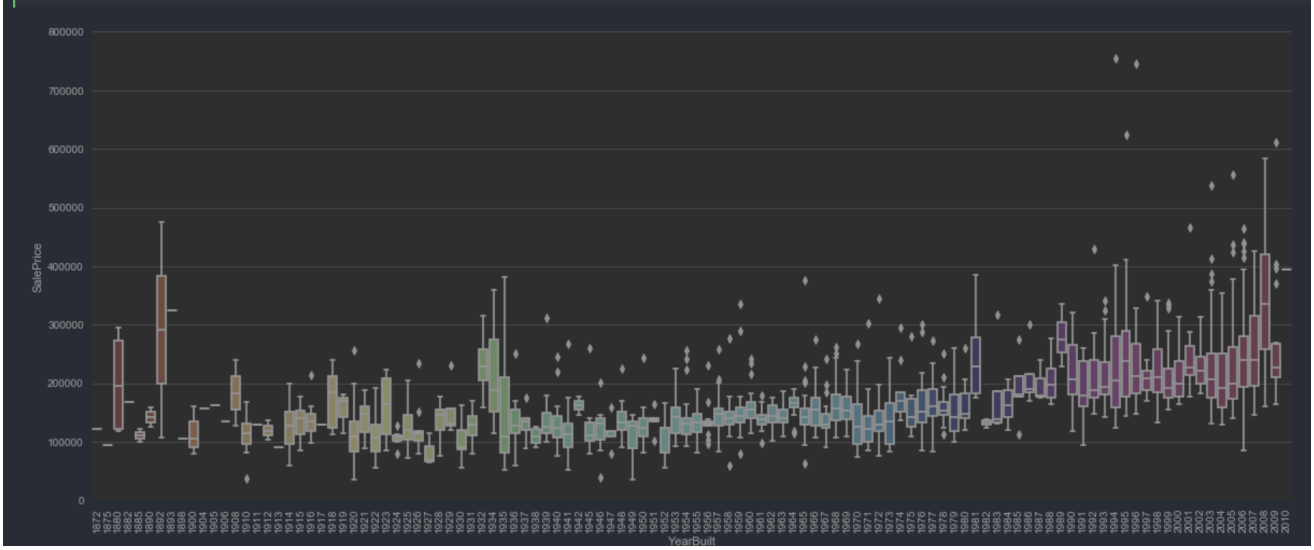
结合ppt和实际的程序运行，总结一下一个基于实际的数据集进行机器学习的过程大致包含哪些关键阶段？每个阶段可以进行的操作都有哪些？为什么要进行这些操作？可以结合实际的运行结果来说明。

观察数据

读入训练集，获取统计特征，根据直方图等可视化工具、偏度和峰度、相关矩阵等统计指标，对数据有初步的了解，或发现数据异常。

如实验程序中1.9，通过箱线图观察 `YearBuilt` 和 `SalePrice` 的关系。

```
# P1.9 YearBuilt与SalePrice关系
var = 'YearBuilt'
data = pd.concat([df_train['SalePrice'], df_train[var]], axis=1)
f, ax = plt.subplots(figsize=(20, 8))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);
plt.xticks(rotation=90);
```



直观地可以看出，同一年代的房屋，房价近似正态分布。以50年为窗口期，平均波动并不大。

异常数据统计和处理

- 离群值：可以进行清洗或删除。如：对数转换、缩尾、截尾、插值。
- 缺失值：一般有三种处理方式：缺失比例大者（如15%）删除列；删除所在行；进行填充（可用前一个非缺失值填充。对于其他的分类特征，填none；对于数值特征，用中位数、众数等）。

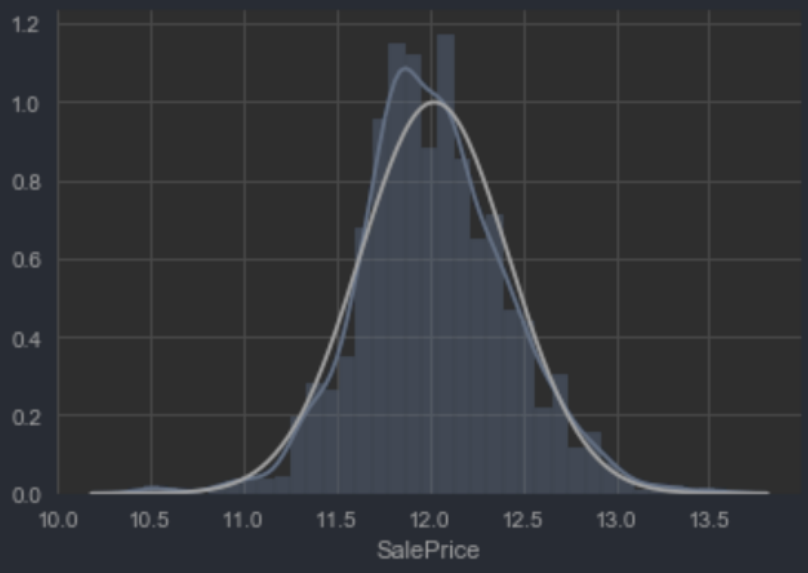
处理之后，可以再次观察数据，可能需要继续清洗、选择。

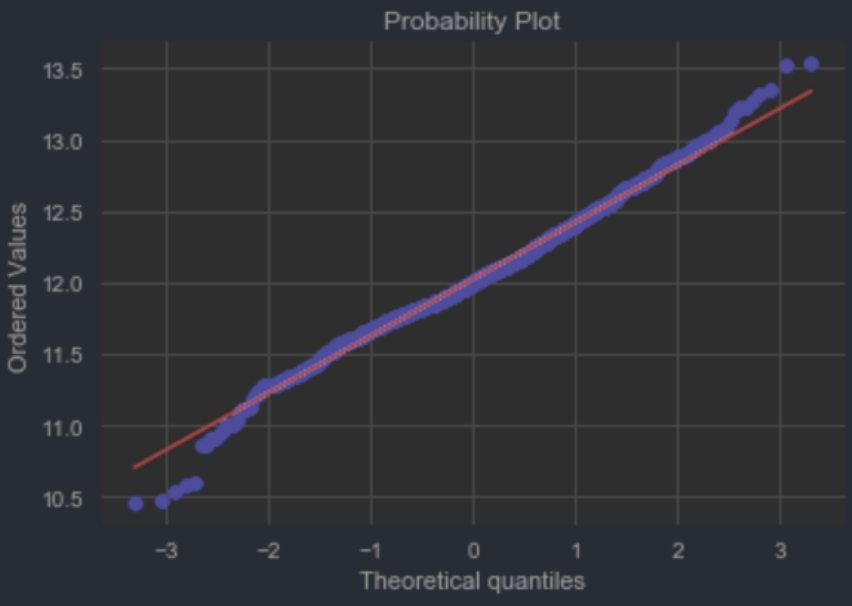
特征工程

根据相关矩阵、最相关矩阵、相关系数等数学模型，针对数值特征进行相关性分析，**进行特征选择和特征表示**。比如：

- 对非数值型列，需要进行独热编码等。
- 处理高相关性的特征，避免共线性导致模型失真。
- 对数值型特征进行正态分布检查，必要时做对数变换，实现同方差。如下图，对应的”尖峰“、”右偏“进行了矫正。

```python
# P1.17 对数变换
df_train['SalePrice'] = np.log1p(df_train['SalePrice'])
# 绘制调整后的直方图与概率图
sns.distplot(df_train['SalePrice'], fit=norm);
fig = plt.figure()
res = stats.probplot(df_train['SalePrice'], plot=plt)
```





特征值处理和调整

根据可视化的结果，进行模型处理。如：了解样本的分布，以优化样本值；进行标准化。

建立模型和训练

划分出合适的数据集和训练集后，可以对模型进行训练，预期获得较低的损失函数。如使用岭回归模型、Lasso回归模型、XGBoost回归模型等进行预测。

模型评估与比较

通过MSE等指标，评估模型的表现，考虑模型的泛化能力等，选择符合自己需求的模型。

特征工程的重要性

> 其中有一个关键的阶段是进行特征选择和特征表示。请问如果不进行特征选择会怎样？

如果我们拥有大量特征时，需要判断哪些是相关特征、哪些是不相关特征，选取合适的特征进行模型迭代，既兼顾代表性、可解释性又不造成维数灾难。

神经网络探索

> 对于多层神经网络模型，尝试通过增加模型复杂度、过度训练让其达到过拟合的效果，给出在非过拟合和过拟合情形下的模型描述（包括层数、节点数、激活函数类型等）、训练次数、以及在训练集、验证集的均方根误差。

调参剪影图：

[225]:
```
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error

model = MLPRegressor(hidden_layer_sizes=(1000), activation="relu",
                     solver='adam', alpha=0.01,
                     batch_size='auto', learning_rate="constant",
                     learning_rate_init=0.1, power_t=0.5, max_iter=20000)
model.fit(x_train,y_train)
print(model.score(x_val, y_val))
print(mean_squared_error(model.predict(x_train), y_train))
print(mean_squared_error(model.predict(x_val), y_val))
```

```
/home/ll/miniconda3/envs/ML/lib/python3.8/site-packages/sklearn/neural_n
ng: A column-vector y was passed when a 1d array was expected. Please ch
vel().
  y = column_or_1d(y, warn=True)
0.576828958743083
0.03862983036708511
0.06400660401436502
```

	层数	节点数	激活函数类型	训练次数	训练集均方根误差	验证集均方根误差
非过拟合	5	5	RELU	6666	0.21	0.20
过拟合	648	648	RELU	114514	0.02	0.31