第一次作业

请复习《机器学习实战》教材的第 2“端到端的机器学习项目”，仔细阅读该章节的配套源码“02_end_to_end_machine_learning_project.ipynb”，并仿照该源码，尝试写一段 python 程序，利用 SVM 来解决房价预测问题。并完成以下问答题。

1、 问题设定。通过沟通确定了业务目标，明确了要解决的问题属于回归问题，并选择了相应的性能衡量指标。本章中，所使用的性能衡量指标是什么？

所使用的性能衡量指标是：RMSE（均方根误差）

2、 获取数据。假定数据集已下载到本地，

（1） 如何加载数据集？请写出相应的代码。

```
import pandas as pd
def load_housing_data(housing_path=HOUSING_PATH):
    csv_path = os.path.join(housing_path, "housing.csv")
    return pd.read_csv(csv_path)
housing = load_housing_data()
```

（2） 数据集的总行数是？每个属性的类型是？哪个属性存在缺失值？

数据集总行数：20640

每个属性对应的类型如下：

```
Column              Dtype
------              -----
longitude           float64
latitude            float64
housing_median_age  float64
total_rooms         float64
total_bedrooms      float64
population          float64
households          float64
median_income       float64
median_house_value  float64
ocean_proximity     object
```

存在缺失值的属性：total_bedrooms

（3） 数据集中的 ocean_proximity 属性为非数值属性，其取值情况有哪几种？

ocean_proximity 属性的取值有以下 5 种：

```
<1H OCEAN     9136
INLAND        6551
NEAR OCEAN    2658
NEAR BAY      2290
ISLAND           5
```

（4） 如何查看数据集中所有数值型属性的平均值，最大值和最小值。

运行：housing.describe()

表格中对应的 mean、max、min 行就是数据集中各项数据的平均值、最大值、

最小值。

（5） 如何查看数据集中所有数值型属性的直方图？观察直方图，如何判断某个属性的取值被设定了上限？直方图中的“重尾”是指什么？发现重尾现象，需要进行处理吗？

在数据集上调用 hist()方法，即可绘制每个属性的直方图：housing.hist(bins=50, figsize=(20,15))

在直方图中，若明显可以看出某个属性的最大值超过了正常水平，即在直方图中表现为最大值的纵轴过高，则表示该属性可能被设定了上限。

直方图中的“重尾”指的是：图形在中位数右侧的延伸比左侧要远得多。

发现重尾现象时需要进行转换处理。

（6） 创建测试集时，通常有哪两类抽样方式？本章中哪种抽样方式更好？为什么？

纯随机抽样和分层抽样

本章中分层抽样更好，因为分层抽样时测试集能够代表整个数据集中各个不同类型的特征，分层抽样的测试集中的比例分布与完整数据集中的分布几乎一致。

3、 研究数据。

（1） 我们需要对测试集中的样本进行研究吗？为什么？

不需要，因为测试集的作用是用来判断训练出的模型的准确率的，若对测试集数据进行研究，可能会产生数据窥探偏误。

（2） 通过探寻属性之间的相关性，可以找出最重要的特征。有哪些函数可以用于探寻属性之间的相关性？

corr()函数可以计算每对属性之间的标准相关系数（皮尔逊相关系数）；

scatter_matrix()函数可以绘制出每个数值属性相对于其他数值属性的相关性。

（3） 尝试增加“每个家庭的房间数量”、“每个家庭的人口数”、“卧室/房间比例”三个属性。与房价中位数相关性最高的 4 个属性分别是？

median_income

bedrooms_per_room

rooms_per_household

latitude

4、 准备数据。

（1） 在此环节，需要对测试集中的数据做相关处理吗？

不需要。

（2） 你是如何处理缺失值的？

用缺失值属性的中位数值替换该属性的缺失值。

（3） 为什么必须将 ocean_proximity 属性的文本类型转换为数值类型的？

因为机器学习算法更容易处理数值类型，为了让数据适应算法和库，需要将文本类型转换为数值类型。

（4） 你是如何将 ocean_proximity 属性的文本类型转换为数值类型的？

先调用 OridinalEncoder()将文本标签转化为数字，再使用 OneHotEncoder 编码器，将整数分类值转化为独热向量。

（5） 尝试增加 3 个新的重要特征：“每个家庭的房间数量”、“每个家庭的人口数”、“卧室/房间比例”

```
from sklearn.base import BaseEstimator, TransformerMixin
rooms_ix, bedrooms_ix, population_ix, households_ix = 3, 4, 5, 6
class CombinedAttributesAdder(BaseEstimator, TransformerMixin):
    def __init__(self, add_bedrooms_per_room=True):
        self.add_bedrooms_per_room = add_bedrooms_per_room
    def fit(self, X, y=None):
        return self    # nothing else to do
    def transform(self, X):
        rooms_per_household = X[:, rooms_ix] / X[:, households_ix]
        population_per_household = X[:, population_ix] / X[:, households_ix]
        if self.add_bedrooms_per_room:
            bedrooms_per_room = X[:, bedrooms_ix] / X[:, rooms_ix]
            return np.c_[X, rooms_per_household, population_per_household,
                         bedrooms_per_room]
        else:
            return np.c_[X, rooms_per_household, population_per_household]
```

调用 transform 函数后即可添加三个新的特征。

（6） 你用到了特征缩放吗？你是如何实现特征缩放的？

用到了特征缩放，通过在转换流水线中设置标准化转换器 StandardScaler 来实现特征缩放。

（7） 尝试使用流水线，来对数据做相关处理，包括缺失值处理，文本类型转换为数值类型，特征缩放，增加新的重要属性。（这是对提交的 Python 代码的要求）

5、研究模型。

（1） 训练一个 SVM（使用线性核函数）。并在训练集上评估其性能。在训练集上的 RMSE 是？

在训练集上的 RMSE 是：111094.6308539982

（2） 利用 10 折交叉验证来评估其泛化性能。在验证集上的 RMSE 均值是？

验证集上的 RMSE 均值是：111809.84009600841

6、微调模型。

（1） 选择最佳超参。

a) 利用网格搜索，从以下超参组合中选取最佳超参。其中，网格搜索的配置为“cv=5, scoring='neg_mean_squared_error', verbose=2”。最后得到的最佳超参是？最佳超参时，验证集上的 RMSE 为多少？

```
    param_grid = [
            {'kernel': ['linear'], 'C': [10., 30., 100., 300., 1000., 3000., 10000., 30000.0]},
            {'kernel': ['rbf'], 'C': [1.0, 3.0, 10., 30., 100., 300., 1000.0],
             'gamma': [0.01, 0.03, 0.1, 0.3, 1.0, 3.0]},
        ]
```

最后得到的最佳超参是：{'C': 30000.0, 'kernel': 'linear'}

最佳超参时，验证集上的 RMSE 为：70363.84006944533

b) **利用随机搜索，寻找超参。**

随机搜索的配置如下：“param_distributions=param_distribs, n_iter=50, cv=5, scoring='neg_mean_squared_error', verbose=2, random_state=42) ”，其中：param_distribs = {

```
'kernel': ['linear', 'rbf'],
'C': reciprocal(20, 200000),
'gamma': expon(scale=1.0),
}
```

最后得到的最佳超参是：

{'C': 157055.10989448498, 'gamma': 0.26497040005002437, 'kernel': 'rbf'}

最佳超参时，验证集上的 RMSE 为：54767.960710084146

（2）在测试集上评估系统。其在测试集上的 RMSE 是？（提示：利用流水线的 transform() 方法对测试集做相关处理后，再进行评估）

用随机搜索获得的最佳超参，在测试集上评估系统，其 RMSE 是：52490.03793898214