实战案例5：贷款审批结果预测

作者： Robin
日期： 2018/09
数据集来源： kaggle

1. 案例描述

该项目的目的主要是基于贷款记录的各项信息预测该笔贷款是否有风险。

2. 数据集描述

- Kaggle提供的数据集包含1000条贷款记录，每条记录包含9项信息。
- 数据字典
  - **Age**: 年龄，整型
  - **Sex**: 性别，字符串（male, female）
  - **Job**: 工作等级，整型，0-4表示从事“非技能”到“技能”的工种，值越大表示从事越高的技能工种
  - **Housing**: 住房状况，字符串，free：无住房，rent：租房，own：拥有住房
  - **Saving accounts**: 储蓄账户，字符串，little：很少, moderate：中等, quite rich：较多, rich：很多
  - **Checking account**: 支票账户，字符串，little：很少, moderate：中等, rich：很多
  - **Credit amount**: 信用卡可用额度，整型
  - **Duration**: 贷款期限，整型，单位（月）
  - **Purpose**: 贷款目的，字符串
  - **Risk**: 贷款质量（是否违约），字符串，good: 没有违约，bad：违约

3. 任务描述

- 使用scikit-learn建立不同的机器学习模型进行贷款违约预测

4. 主要代码解释

- 代码结构

```
lect05_proj
├── data
    ├── german_credit_data.csv    # CSV数据文件
├── output
    ├── model_comparison.csv    # 模型比较结果的CSV文件(程序的输出)
    ├── pred_results.png    # 模型比较结果的png文件(程序的输出)
├── config.py     # 配置文件
├── utils.py     # 工具类文件
├── main.py     # 主程序

├── lect05_proj_readme.pdf  # 案例讲解文档
```

- **utils.py**

使用LabelEncoder()首先对类别特征进行标签编码转换为数字标签，由于LabelEncoder()只能对一列数据做编码，所以这里需要循环遍历所有的类别特征列。

```
def transform_train_data(train_data):
    ...
    # 类别型数据
    label_feats = None
    for cat_col in config.cat_cols:
        label_enc = LabelEncoder()
        label_feat = label_enc.fit_transform(train_data[cat_col].values).reshape(-1, 1)
    ...
```

- **utils.py**

使用OneHotEncoder()进行特征的独热编码，sparse=False表明返回的结果不是按照稀疏矩阵返回的

```
def transform_train_data(train_data):
    ...
    # 独热编码处理类别特征
    onehot_enc = OneHotEncoder(sparse=False)
    ...
    onehot_feats = onehot_enc.fit_transform(label_feats)
    ...
```

- **utils.py**

使用MinMaxScaler()进行范围归一化（默认归一化后的范围为[0, 1]）

```
def transform_train_data(train_data):
    ...
    scaler = MinMaxScaler(feature_range=(0, 1))
    ...
    scaled_all_feats = scaler.fit_transform(all_feats)
    ...
```

- **utils.py**

使用PCA()进行特征降维，参数n_components为整数时，表示降维后的主成分个数；参数n_components为浮点数时，表示按照“贡献率”进行主成分选取

```
def transform_train_data(train_data):
    ...
    # 使用特征降维
    pca = PCA(n_components=0.99)
    X_train = pca.fit_transform(scaled_all_feats)
    ...
```

- **utils.py**

对测试数据的特征进行处理时，使用来自训练数据的标签encoder，独热编码encoder，范围归一化scaler及降维pca

```
def transform_test_data(test_data, label_encs, onehot_enc, scaler, pca):
    ...
```

- **main.py**

使用的模型及相关参数配置。该项目中使用了8个机器学习模型，并为不同的学习模型指定了参数空间。 如：kNN，指定了3个k值用于比较对结果的影响：5, 25, 55

```
def main():
    ...
    model_name_param_dict = {'kNN': (KNeighborsClassifier(),
                                     {'n_neighbors': [5, 25, 55]}),
                             'LR': (LogisticRegression(),
                                    {'C': [0.01, 1, 100]}),
                             'SVM': (SVC(probability=True),
                                     {'C': [0.01, 1, 100]}),
                             'DT': (DecisionTreeClassifier(),
                                    {'max_depth': [50, 100, 150]}),
                             'Stacking': (sclf,
                                          {'kneighborsclassifier__n_neighbors': [5, 25, 55],
                                           'svc__C': [0.01, 1, 100],
                                           'decisiontreeclassifier__max_depth': [50, 100, 150],
                                           'meta-logisticregression__C': [0.01, 1, 100]}),
                             'AdaBoost': (AdaBoostClassifier(),
                                          {'n_estimators': [50, 100, 150, 200]}),
                             'GBDT': (GradientBoostingClassifier(),
                                      {'learning_rate': [0.01, 0.1, 1, 10, 100]}),
                             'RF': (RandomForestClassifier(),
                                    {'n_estimators': [100, 150, 200, 250]})}
    ...
```

- **utils.py**

GridSearchCV()进行网格搜索和交叉验证。注意，由于数据不均衡，所以这里使用AUC值作为评价指标。

```
def train_test_model(X_train, y_train, X_test, y_test, model_name, model, param_range):
    ...
    clf = GridSearchCV(estimator=model,
                       param_grid=param_range,
                       cv=5,
                       scoring='roc_auc',
                       refit=True)
    ...
```

5. 案例总结

- 该项目通过预测贷款是否可能违约，巩固并实践了使用scikit-learn搭建简单的预测模型:
    - 机器学习流程
    - scikit-learn中常用预测模型的使用，如：kNN, logistic regression，decision tree及SVM
    - 参数的选择

6. 课后练习

- 参考随堂代码，试着使用不同的参数空间观察对结果的影响。

参考资料

1. scikit-learn官方教程
2. 通过scikit-learn理解机器学习