题目　Real-World Data Mining Competition.

姓名 骆克云 学号 MG1633052 邮箱 streamer.ky@foxmail.com 联系方式 18115128082

(南京大学 计算机科学与技术系，南京　210093)

1　实现细节

1.1 数据预处理

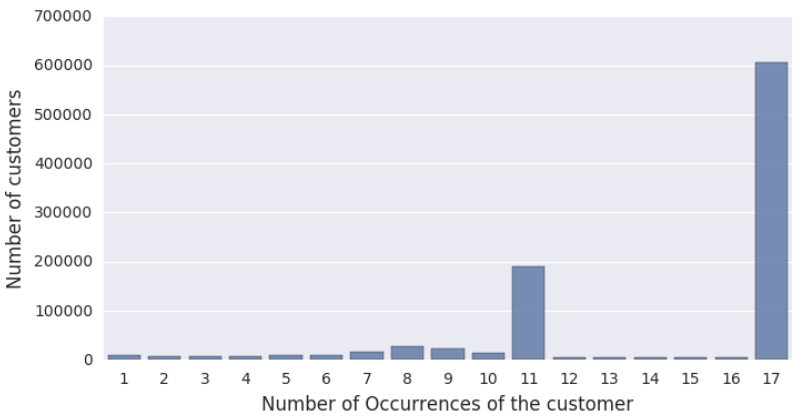
读取训练集和测试集，填补缺失数据，特征映射，划分 train_X,train_y,test_X, test_y，写入 CSV 文件。

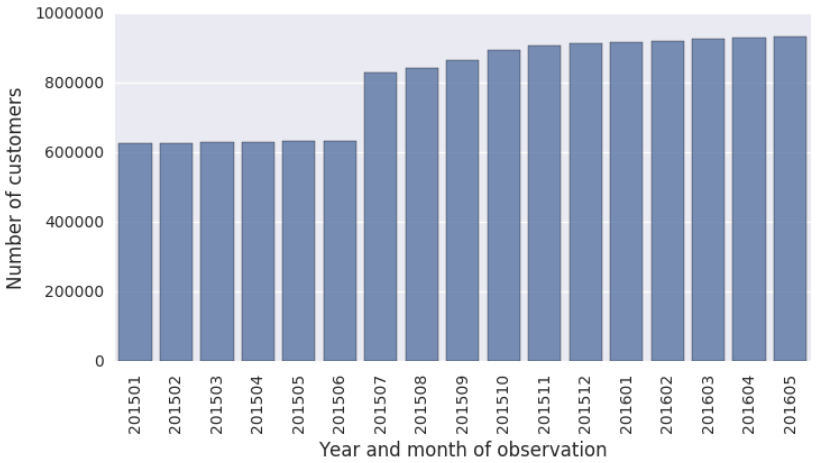
本方法首先使用 pandas 读取原始数据，然后对一些缺失值进行填充，针对离散特征主要填充 0，对于连续特征（'age'，'renta'，'antiguedad'）填充该特征的平均值。

经过该阶段，生成 data_train_clean.csv， train_file_clean.csv（201501-201506）； data_test_clean.csv，test_file_clean.csv（201601-201606），供后续生成测试集与训练集。

1.2 特征选择与分析

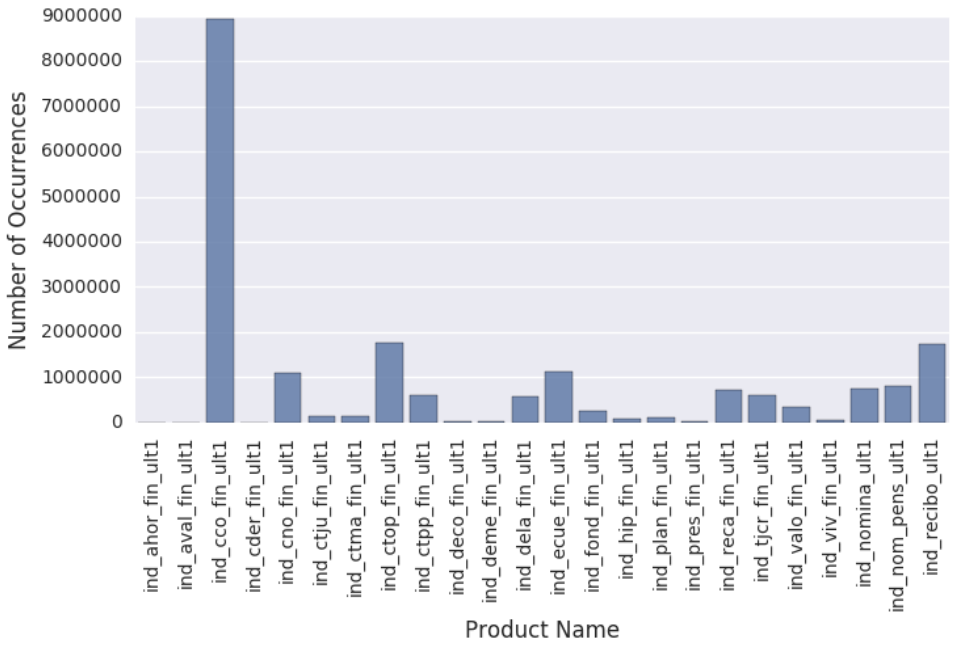
Santander 数据共有 24 个特征，24 个预测值。不同用户的活跃度不同，但大部分用户都有 17 个月的数据，





并且前 6 个月的用户数据偏少，后面 11 个月用户数基本相当。

预测值量化特征如下，可以发现有部分商品购买量很少，因此预测时可以考虑去除这些商品。



参照论坛上 When Less is More 思想，使用 XGBoost 工具编写代码调参提交。

经过不断测试提交，最终选定的单个特征如下：['sexo', 'segmento','ind_empleado', 'canal_entrada', 'nomprov', 'ind_nuevo', 'pais_residencia', 'ind_actividad_cliente', 'age', 'renta', 'antiguedad']，共 11 个。

预测的项目如下(排除了 4 个): ['ind_cco_fin_ult1','ind_cder_fin_ult1', 'ind_cno_fin_ult1', 'ind_ctju_fin_ult1', 'ind_ctma_fin_ult1', 'ind_ctop_fin_ult1', 'ind_ctpp_fin_ult1', 'ind_dela_fin_ult1', 'ind_ecue_fin_ult1', 'ind_fond_fin_ult1', 'ind_hip_fin_ult1', 'ind_plan_fin_ult1', 'ind_pres_fin_ult1', 'ind_reca_fin_ult1', 'ind_tjcr_fin_ult1', 'ind_valo_fin_ult1', 'ind_viv_fin_ult1', 'ind_nomina_ult1', 'ind_nom_pens_ult1', 'ind_recibo_ult1'], 共 20 个。

实际运行中发现上述特征得分并不是很高，因而又使用了一些复合特征，具体就是将每年前 5 个月每个月

购买的商品作为用户特征添加进去，还加上前 5 个月所有购买的商品特征。

1.3 生成训练集与测试集

1) 训练集：2015 年 1 月至 6 月数据，其中前五个月为 X 数值，最后一个月为 y 数值，具体来说先读取'data_train_clean.csv'按月获取用户购买产品信息，然后读取'train_file_clean.csv'将单个特征融合用户每个月购买特征与前五个月购买总的商品种类特征，形成复合特征，预测值则为 6 月份新购买的商品。

```
def get_data_train(file_name, train_file):
    train_X = []
    train_y = []
    user_dict = [{} for _ in range(6)]
    target_len = len(target_cols)

    with open(file_name) as f:
        f_csv = csv.DictReader(f)
        for row in f_csv:
            # 用户特征
            user_id = int(row['ncodpers'])
            i = int(row["fecha_dato"][6]) % 6
            if i != 0:
                target_list = [int(float(row[target])) for target in target_cols]
                user_dict[i][user_id] = target_list[:]

    with open(train_file) as f:
        f_csv = csv.DictReader(f)
        for row in f_csv:
            user_id = int(row['ncodpers'])
            # 特征提取
            X_feature = []
            # 离散特征
            X_feature.append([int(row[col]) for col in cat_cols])
            # 连续特征
            X_feature.append([int(float(row[col])) for col in num_cols])
            X_feature = flattern(X_feature)

            if row['fecha_dato'] == '2015-06-28':
                user05 = user_dict[5].get(user_id, [0] * target_len)
                user01 = user_dict[1].get(user_id, [0] * target_len)
                user02 = user_dict[2].get(user_id, [0] * target_len)
                user03 = user_dict[3].get(user_id, [0] * target_len)
                user04 = user_dict[4].get(user_id, [0] * target_len)
                already_had = had_in_past(user05, user04, user03, user02, user01)
                user06 = [int(float(row[target])) for target in target_cols]
```

```
                new_products = [max(x6 - x5, 0) for (x6, x5) in zip(user06, user05)]
                # 仅 6 月份购买过商品的用户参与训练
                if sum(new_products) > 0:
                    for ind, prod in enumerate(new_products):
                        if prod > 0:
                            # assert len(user05) == target_len
                            train_X.append(
                                X_feature + user05 + user04 + user02 + user01 + user03 +
already_had)
                            train_y.append(ind)
    return np.array(train_X, dtype=int), np.array(train_y, dtype=int)
```

2) 测试集：同上，但是读取的是 2016 年的数据，y 预测值是缺失的。

1.4 使用XGBoost进行训练和预测

XGBoost 模型如下：

```
def XGBModel(train_X, train_y, seed=2016):
    param = {}
    param['objective'] = 'multi:softprob'
    param['eta'] = 0.1
    param['max_depth'] = 9
    param['silent'] = 1
    param['num_class'] = 20
    param['eval_metric'] = "mlogloss"
    param['min_child_weight'] = 4
    param['gamma'] = 3
    param['subsample'] = 0.90
    param['colsample_bytree'] = 0.9
    param['seed'] = seed
    num_rounds = 80

    plst = list(param.items())
    xgtrain = xgb.DMatrix(train_X, label=train_y)
    model = xgb.train(plst, xgtrain, num_rounds)
    return model
```

经过不断调参获得如上最佳效果。使用多类概率分类，树的最大深度为 9，步长 0.1，分类数为 20，损失函数为多类 log 损失函数，孩子节点剪枝阈值为 4，gamma 阈值为 2，样本采样数和特征采样数均为 0.9，每一次运行 80 轮。

1.5 结果运行写入

1）清洗数据(第一次运行)。

2）获取训练集和测试集。

3）训练模型，进行预测，得到预测值。

4）在预测值中，去掉 2016 年 5 月份已经出现的产品(预测不是完全可靠的，会预测出 5 月已有的产品)，取前 7 个概率最大且不为 0 的值对应的索引，若不足 7 个则取尽可能多的。

5）取测试集 ID，将预测结果对应的名称拼接在一起，得到最终结果。

6）写到 CSV 文件中。

2 结果

2.1 实验设置

数据来源：train_ver2.csv，test_ver2.csv

数据预处理：训练数据由 20150128~20150528：data_train_clean.csv， 20150628：train_file_clean.csv 组成;测试数据由 20160128~20160528：data_test_clean.csv， 20160628：test_file_clean.csv 组成。

2.2 实验结果

1. Kaggle 排名：

   A 榜最高得分为：0.0301044，90 名左右。

2. 结果反思

选取好的特征往往事半功倍，特征工程很重要。

注意：

1. 最终提交的报告最好保存为 pdf 格式
2. 压缩格式为 zip 格式，请勿使用需要安装特定软件才能打开的压缩方式
3. 作业的文件夹目录请按照网页要求，代码、结果放在不同子文件夹中。作业网页上给出的数据不需要再次提交