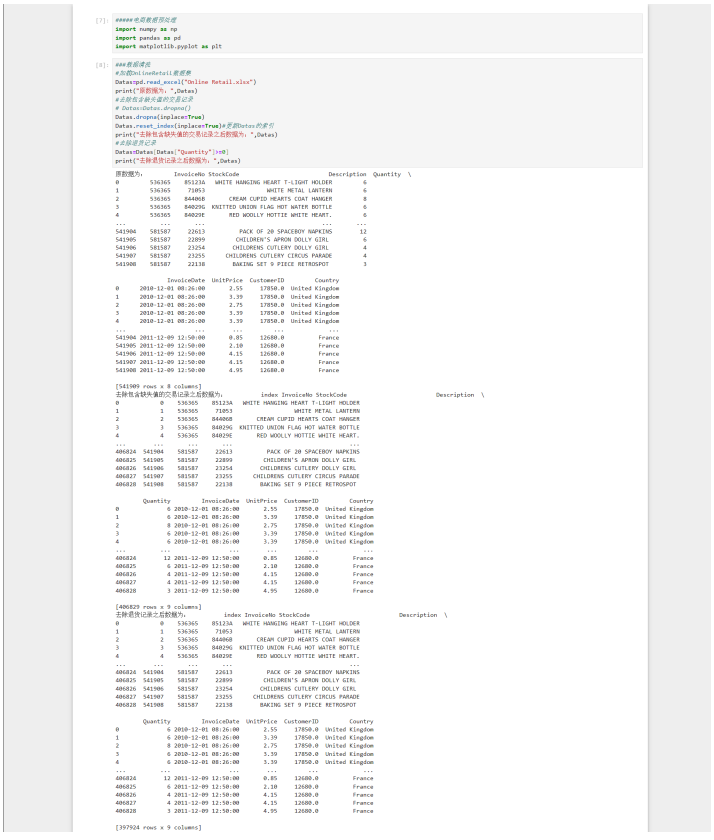
第十四周问答作业

朱士杭 231300027

2024 年 5 月 29 日

1 问题一：电商数据预处理

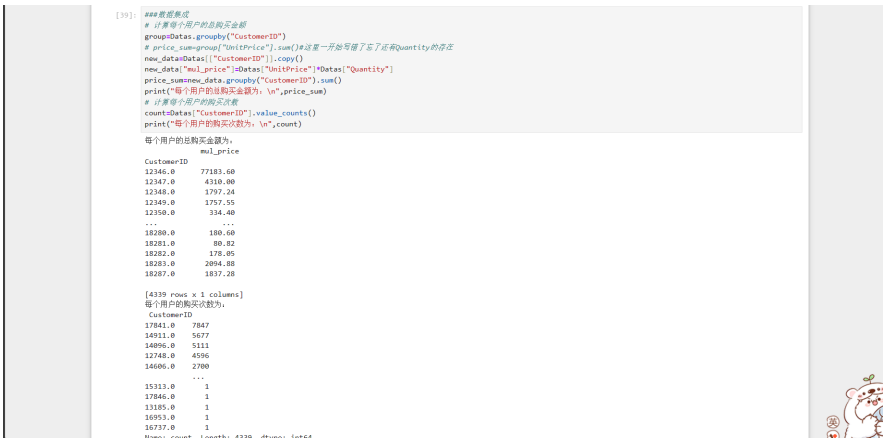
具体见“第十四周问答作业.ipynb”



```
[7]: #####电商数据预处理
     import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt

[8]: ###数据读取
     #当前Online Retail.excel数据集
     Datas=pd.read_excel("Online Retail.xlsx")
     print("原数据为：",Datas)
     #去除在之前生成的交易记录
     # Datas=Datas.dropna()
     Datas.dropna(inplace=True)
     Datas.reset_index(inplace=True)#更新Datas的索引
     print("去除包含缺失值的交易记录之后的版为：",Datas)
     #去除退货记录
     Datas=Datas[Datas['Quantity']>0]
     print("去除退货记录之后的版为：",Datas)
```

```
原数据为：       InvoiceNo StockCode                          Description  Quantity  \
0         536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER         6
1         536365     71053                  WHITE METAL LANTERN         6
2         536365    84406B       CREAM CUPID HEARTS COAT HANGER         8
3         536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE         6
4         536365    84029E       RED WOOLLY HOTTIE WHITE HEART.         6
...          ...       ...                                  ...       ...
541904    581587     22613          PACK OF 20 SPACEBOY NAPKINS        12
541905    581587     22899         CHILDREN'S APRON DOLLY GIRL          6
541906    581587     23254       CHILDRENS CUTLERY DOLLY GIRL           4
541907    581587     23255     CHILDRENS CUTLERY CIRCUS PARADE          4
541908    581587     22138       BAKING SET 9 PIECE RETROSPOT           3

            InvoiceDate  UnitPrice  CustomerID         Country
0      2010-12-01 08:26:00       2.55     17850.0  United Kingdom
1      2010-12-01 08:26:00       3.39     17850.0  United Kingdom
2      2010-12-01 08:26:00       2.75     17850.0  United Kingdom
3      2010-12-01 08:26:00       3.39     17850.0  United Kingdom
4      2010-12-01 08:26:00       3.39     17850.0  United Kingdom
...                    ...        ...         ...             ...
541904 2011-12-09 12:50:00       0.85     12680.0          France
541905 2011-12-09 12:50:00       2.10     12680.0          France
541906 2011-12-09 12:50:00       4.15     12680.0          France
541907 2011-12-09 12:50:00       4.15     12680.0          France
541908 2011-12-09 12:50:00       4.95     12680.0          France

[541909 rows x 8 columns]
去除包含缺失值的交易记录之后的版为：        index InvoiceNo StockCode                          Description  \
0           0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER
1           1    536365     71053                  WHITE METAL LANTERN
2           2    536365    84406B       CREAM CUPID HEARTS COAT HANGER
3           3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE
4           4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.
...       ...       ...       ...                                  ...
406824 541904    581587     22613          PACK OF 20 SPACEBOY NAPKINS
406825 541905    581587     22899         CHILDREN'S APRON DOLLY GIRL
406826 541906    581587     23254       CHILDRENS CUTLERY DOLLY GIRL
406827 541907    581587     23255     CHILDRENS CUTLERY CIRCUS PARADE
406828 541908    581587     22138       BAKING SET 9 PIECE RETROSPOT

       Quantity      InvoiceDate  UnitPrice  CustomerID         Country
0             6  2010-12-01 08:26:00       2.55     17850.0  United Kingdom
1             6  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
2             8  2010-12-01 08:26:00       2.75     17850.0  United Kingdom
3             6  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
4             6  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
...         ...                  ...        ...         ...             ...
406824       12  2011-12-09 12:50:00       0.85     12680.0          France
406825        6  2011-12-09 12:50:00       2.10     12680.0          France
406826        4  2011-12-09 12:50:00       4.15     12680.0          France
406827        4  2011-12-09 12:50:00       4.15     12680.0          France
406828        3  2011-12-09 12:50:00       4.95     12680.0          France

[406829 rows x 9 columns]
去除退货记录之后的版为：        index InvoiceNo StockCode                          Description  \
0           0    536365    85123A   WHITE HANGING HEART T-LIGHT HOLDER
1           1    536365     71053                  WHITE METAL LANTERN
2           2    536365    84406B       CREAM CUPID HEARTS COAT HANGER
3           3    536365    84029G  KNITTED UNION FLAG HOT WATER BOTTLE
4           4    536365    84029E       RED WOOLLY HOTTIE WHITE HEART.
...       ...       ...       ...                                  ...
406824 541906    581587     22613          PACK OF 20 SPACEBOY NAPKINS
406825 541905    581587     22899         CHILDREN'S APRON DOLLY GIRL
406826 541906    581587     23254       CHILDRENS CUTLERY DOLLY GIRL
406827 541907    581587     23255     CHILDRENS CUTLERY CIRCUS PARADE
406828 541908    581587     22138       BAKING SET 9 PIECE RETROSPOT

       Quantity      InvoiceDate  UnitPrice  CustomerID         Country
0             6  2010-12-01 08:26:00       2.55     17850.0  United Kingdom
1             6  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
2             8  2010-12-01 08:26:00       2.75     17850.0  United Kingdom
3             6  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
4             6  2010-12-01 08:26:00       3.39     17850.0  United Kingdom
...         ...                  ...        ...         ...             ...
406824       12  2011-12-09 12:50:00       0.85     12680.0          France
406825        6  2011-12-09 12:50:00       2.10     12680.0          France
406826        4  2011-12-09 12:50:00       4.15     12680.0          France
406827        4  2011-12-09 12:50:00       4.15     12680.0          France
406828        3  2011-12-09 12:50:00       4.95     12680.0          France

[397924 rows x 9 columns]
```

图 1: 电商数据清洗



```
[39]: ###数据集成
      # 计算每个用户的总购买金额
      group=Datas.groupby("CustomerID")
      # price_sum=group["UnitPrice"].sum()#这里一开始写错了忘了还有Quantity的存在
      new_data=Datas[["CustomerID"]].copy()
      new_data["mul_price"]=Datas["UnitPrice"]*Datas["Quantity"]
      price_sum=new_data.groupby("CustomerID").sum()
      print("每个用户的总购买金额为：\n",price_sum)
      # 计算每个用户的购买次数
      count=Datas["CustomerID"].value_counts()
      print("每个用户的购买次数为：\n",count)
```

每个用户的总购买金额为：

CustomerID	mul_price
12346.0	77183.60
12347.0	4310.00
12348.0	1797.24
12349.0	1757.55
12350.0	334.40
...	...
18280.0	180.60
18281.0	80.82
18282.0	178.05
18283.0	2094.88
18287.0	1837.28

[4339 rows x 1 columns]

每个用户的购买次数为：

CustomerID	
17841.0	7847
14911.0	5677
14096.0	5111
12748.0	4596
14606.0	2700
...	
15313.0	1
17846.0	1
13185.0	1
16953.0	1
16737.0	1

Name: count, Length: 4339, dtype: int64

图 2: 电商数据集成

```
[40]: ###数据变换—将用户的总购买金额和购买次数进行标准化处理。
      #将用户的总购买金额归一化，这里统一使用z-score方法
      mean=price_sum["mul_price"].mean()
      std=price_sum["mul_price"].std()
      price_sum["standard_price"]=(price_sum["mul_price"]-mean)/std
      #将用户的购买次数归一化处理
      price_sum=pd.merge(price_sum,count,on="CustomerID")
      mean=price_sum["count"].mean()
      std=price_sum["count"].std()
      price_sum["standard_count"]=(price_sum["count"]-mean)/std
      print("经过归一化处理之后用户的总购买金额与购买次数为：\n",price_sum[["standard_price","standard_count"]])
```

经过归一化处理之后用户的总购买金额与购买次数为：

CustomerID	standard_price	standard_count
12346.0	8.358671	-0.396466
12347.0	0.251017	0.394642
12348.0	-0.028543	-0.265343
12349.0	-0.032959	-0.081771
12350.0	-0.191293	-0.326534
...	...	...
18280.0	-0.208405	-0.357130
18281.0	-0.219506	-0.370242
18282.0	-0.208688	-0.348388
18283.0	0.004571	2.903462
18287.0	-0.024088	-0.094884

[4339 rows x 2 columns]

图 3: 电商数据变换

2 问题二：为什么在 PCA 之前需要进行标准化处理

如果不进行标准化处理的话很有可能出现“大数吃小数”的现象，比如说

$$((100000, 0.5), (111000, 0.55), (99900, 0.32))$$

这三个数据点，其实沿着 y 轴方向相对变化也是很大的，但是根据 PCA 的公式，第二维 (y 轴) 的数据直接被“吃掉”了，而我们实际在比较的时候应当是相对分布而不是绝对分布，因此需要使用归一化的手段使得数据分布更加“公平”。

3 问题三：使用 PCA 对词向量进行降维和可视化

具体见“第十四周问答作业.ipynb”

图 4: PCA 降维和可视化

4 问题四：评估多元线性回归模型的性能

4.1 如何评估性能

核心想法就是看最终的拟合效果如何，比如说我将原先的数据集分成 3 部分——训练集、测试集和验证集

训练集用于生成多元线性回归模型，测试集用于测试模型误差并进行调优，而最终用验证集的数据去评估模型的性能如何，通过将输入数据投喂进去，然后得到最后的结果（标签）

如果是回归任务，则比较和真正结果相差多少（方差）；如果是分类任务，则求出分类的正确率即可

4.2 常用评估指标

4.2.1 拟合优度（Goodness of Fit）

决定系数（R^2）表示模型解释的变异性的比例，值越接近 1 说明模型解释能力越强

调整后的决定系数（Adjusted R^2）考虑到自变量数量的影响，当自变量增多时，R^2 往往会上升，调整后的 R^2 对自变量的数量进行了惩罚，能更准确地反映模型的性能

4.2.2 F 检验

用于检验模型整体是否显著，即至少有一个自变量对因变量有显著影响。

4.2.3 t 检验

对每个回归系数进行检验，判断每个自变量是否对因变量有显著影响。

4.2.4 置信区间

每个回归系数的置信区间，如果置信区间不包括 0，通常认为该系数是显著的

4.2.5 均方误差（MSE）

预测值与实际值之间差异的平方的平均数，MSE 越小，说明模型的预测精度越高。

另一种为均方根误差（RMSE）为 MSE 的平方根，它和因变量的量纲一致，更直观地反映了预测误差的大小。

4.2.6 交叉验证

通过将数据集分成多个部分，在不同的子集上训练和测试模型，来评估模型的稳健性和预测能力。

4.2.7 残差图

检查残差是否随机分布，如果残差图中存在明显模式（如曲线形状或扇形），则说明模型可能存在问题。

另一种是标准化残差，将残差除以其标准差，用于检测异常值和模型的不稳定性。

5 问题五：比较对数几率回归（逻辑斯蒂回归）和 KNN

5.1 对数几率回归（Logistic Regression）

对数几率回归是一种广义线性模型，用于二分类或多分类问题。它通过使用逻辑函数（Logistic Function）将线性回归的输出映射到 0 和 1 之间的概率

对数几率回归关注于找到特征和类别概率之间的最佳拟合, 核心公式是使用一个连续可微的函数 y=1/(1+exp(-wx+b)) 使得一个二分类函数映射到 0-1 之间

训练阶段需要计算参数的梯度并更新，测试阶段对于新的数据点，计算其预测概率是快速的。

通常需要对特征进行缩放，并且可以加入正则化项（如 L1 或 L2 正则化）来防止过拟合

当特征数量较多时，对数几率回归可以有效地进行特征选择和权重分配。它适用于数据量较大，特征维度较高，且需要解释模型的情况

5.2 K-最近邻（KNN）

KNN 是一种基于实例的学习算法，它不需要显式地训练模型参数

预测过程中 KNN 算法会在训练集中找到与待预测数据点最接近的 K 个数据点，并根据这 K 个邻居的类别进行投票来决定待预测点的类别。说白了就是找到跟想要预测的点最相似的 K 个点，再根据他们的类别少数服从多数

KNN 适用于数据量较小，特征维度较低的情况。当数据分布不均匀或存在噪声时，KNN 通过平均多个近邻的响应来减少误差。

5.3 异同点

5.3.1 模型复杂度

对数几率回归在训练阶段有较高的计算复杂度，而 KNN 在测试阶段计算复杂度较高。

5.3.2 数据需求

对数几率回归需要较少的数据来训练模型参数，而 KNN 在训练阶段不需要数据来学习参数，但在预测时需要较多的数据来找到近邻。

5.3.3 泛化能力

对数几率回归通过模型参数来泛化，而 KNN 通过训练数据来泛化。

5.3.4 可解释性

对数几率回归的参数可以直接解释为特征对类别概率的影响，而 KNN 没有显式的参数来解释。

5.3.5 对异常值的敏感性

对数几率回归对异常值相对不敏感，因为它寻找的是全局最优解；而 KNN 对异常值非常敏感，因为异常值会影响近邻的选择。

5.4 适用场景选择

当数据量较大，特征维度较高，且需要模型具有较好的解释性时，优先选择对数几率回归。当数据量较小，特征维度较低，或者数据分布不均匀，噪声较多时，可以考虑使用 KNN

在实际应用中，通常需要根据具体问题和数据集的特点来选择合适的算法，并通过交叉验证等方法来评估和比较不同算法的性能。

6 分类模型对手写数字数据集分类

具体见“第十四周问答作业.ipynb”