Data Processing

Student:hyx

1、Scraping and collect the data

(1)Use pandas’s function ——“read_csv” to read data:

train_df = pd.read_csv('C:/Users/apple/Desktop/dataAnalysis/van_ai_coding_challenge_4-master/van_ai_coding_challenge_4-master/data/train.csv')

(2)Observe the size of the data set：

train_df.shape

2、 Data cleaning and transformation

（1）To understand the specific data form of a dataset, first check the first 10 rows of the dataset：

train_df.head(10)

(2) convert the discrete columns into dummy variables：

dummy_cols = ['Product_ID', 'Gender', 'Age', 'City_Category', 'Stay_In_Current_City_Years']

train_df_with_dummies = pd.get_dummies(train_df, columns= dummy_cols)

(3)check the train_df with dummies:

train_df_with_dummies.head(10)

By now the dimensions still have a good ratio. I didn’t meet dimensionality problem here because my observations are greater than 10 times the number of features.But I still have to be careful not to overfit.

(4)To create X and Y arrays for classification in training phase,I should drop NaN value in the dataframe by using the function dropna() in the Pandas:

train_df_with_dummies_noNaN = train_df_with_dummies.dropna()

```
Console 1/A
In [1]:

In [1]: import sys
   ...: sys.version
   ...: sys.version_info
   ...: # important dependencies
   ...: import os
   ...: import numpy as np
   ...: import pandas as pd
   ...: import matplotlib.pyplot as plt
   ...: import math
   ...: from sklearn.linear_model import ElasticNet, ElasticNetCV, LinearRegression
   ...: from sklearn.ensemble import RandomForestClassifier
   ...: from sklearn.metrics import mean_squared_error
   ...: from sklearn.model_selection import train_test_split
   ...: from scipy import stats

In [2]: train_df = pd.read_csv('C:/Users/apple/Desktop/dataAnalysis/
van_ai_coding_challenge_4-master/van_ai_coding_challenge_4-master/data/train.csv')
   ...: train_df.shape
   ...: # convert the discrete columns into dummy variables
   ...:
   ...: dummy_cols = ['Product_ID', 'Gender', 'Age', 'City_Category',
'Stay_In_Current_City_Years']
   ...: train_df_with_dummies = pd.get_dummies(train_df, columns= dummy_cols)

In [3]: # drop NaN in the dataframe
   ...: train_df_with_dummies_noNaN = train_df_with_dummies.dropna()
   ...:
   ...: train_df_with_dummies_noNaN.head(3)
Out[3]:
    Unnamed: 0  ...  Stay_In_Current_City_Years_4+
0       161273  ...                              0
6       161279  ...                              0
17      161290  ...                              1

[3 rows x 3595 columns]
```