Московский государственный технический университет им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

Лабораторная работа №2

по дисциплине

«Методы машинного обучения»

на тему

Выполнила:

студент Ли Лююй

группы ИУ5И-21М

Москва — 2024 г.

1. Цель лабораторной работы

Цель лабораторной работы: изучение продвинутых способов предварительной обработки данных для дальнейшего формирования моделей.

2. Задание

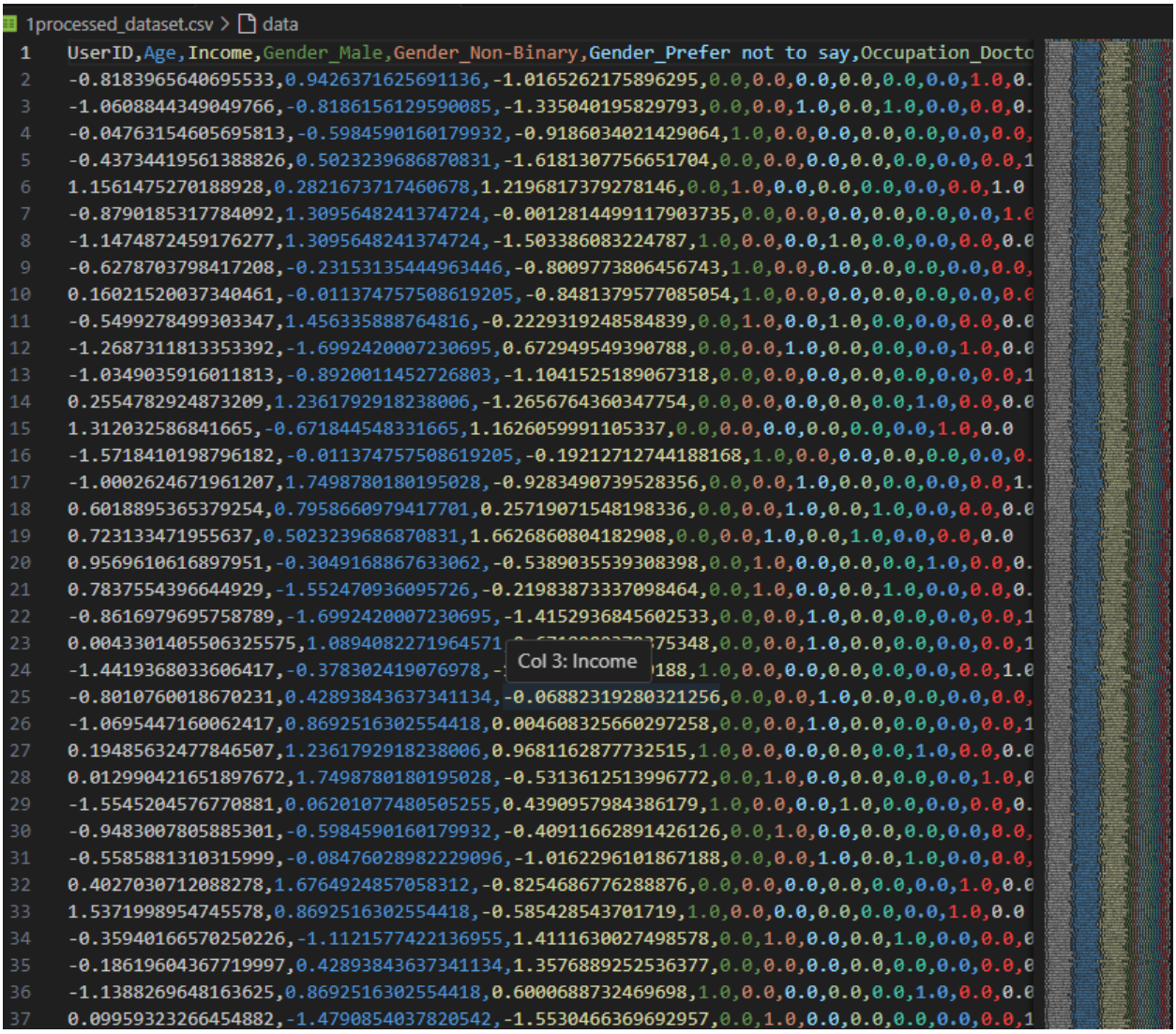
1. Выбрать набор данных (датасет), содержащий категориальные и числовые признаки и пропуски в данных. Для выполнения следующих пунктов можно использовать несколько различных наборов данных (один для обработки пропусков, другой для категориальных признаков и т.д.) Просьба не использовать датасет, на котором данная задача решалась в лекции.
2. Для выбранного датасета (датасетов) на основе материалов лекций решить следующие задачи:
    1.1.1 устранение пропусков в данных;
    1.1.2 кодирование категориальных признаков;
    1.1.3 нормализация числовых признаков.

3. Ход выполнения работы

```
import pandas as pd

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import OneHotEncoder, StandardScaler


# Чтение набора данных

# Убедитесь, что путь к вашему набору данных и имя файла правильные

df = pd.read_csv('generated_dataset.csv')


# Обработка пропущенных значений

# Числовые признаки заполняются медианой

num_imputer = SimpleImputer(strategy='median')

num_columns = df.select_dtypes(include=['int64', 'float64']).columns

df[num_columns] = num_imputer.fit_transform(df[num_columns])
```

```
# Категориальные признаки заполняются наиболее частым значением
cat_imputer = SimpleImputer(strategy='most_frequent')
cat_columns = df.select_dtypes(include=['object']).columns
df[cat_columns] = cat_imputer.fit_transform(df[cat_columns])

# Кодирование категориальных признаков
encoder = OneHotEncoder(drop='first')
encoded_columns = encoder.fit_transform(df[cat_columns]).toarray()
encoded_df = pd.DataFrame(encoded_columns,
columns=encoder.get_feature_names_out(cat_columns))

# Объединение закодированных категориальных признаков с исходным набором данных и удаление исходных категориальных столбцов
df = df.drop(cat_columns, axis=1)
df = pd.concat([df, encoded_df], axis=1)
# Стандартизация числовых признаков
scaler = StandardScaler()
df[num_columns] = scaler.fit_transform(df[num_columns])

# Экспорт обработанных данных в CSV
processed_dataset_path = 'processed_dataset.csv'
df.to_csv(processed_dataset_path, index=False)

print(f"数据预处理完成，文件 '{processed_dataset_path}' 已保存.")
```

Результат:




```
UserID,Age,Income,Gender_Male,Gender_Non-Binary,Gender_Prefer not to say,Occupation_Docto
-0.8183965640695533,0.9426371625691136,-1.0165262175896295,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.
-1.0608844349049766,-0.8186156129590085,-1.335040195829793,0.0,0.0,1.0,0.0,1.0,0.0,0.0,0.
-0.0476315460569581 3,-0.5984590160179932,-0.9186034021429064,1.0,0.0,0.0,0.0,0.0,0.0,0.0,
-0.43734419561388826,0.5023239686870831,-1.6181307756651704,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1
1.1561475270188928,0.2821673717460678,1.2196817379278146,0.0,1.0,0.0,0.0,0.0,0.0,0.0,1.0
-0.8790185317784092,1.3095648241374724,-0.0012814499117903735,0.0,0.0,0.0,0.0,0.0,0.0,1.0
-1.1474872459176277,1.3095648241374724,-1.503386083224787,1.0,0.0,0.0,1.0,0.0,0.0,0.0,0.0
-0.6278703798417208,-0.23153135444963446,-0.8009773806456743,1.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.16021520037340461,-0.011374757508619205,-0.8481379577085054,1.0,0.0,0.0,0.0,0.0,0.0,0.0
-0.5499278499303347,1.456335888764816,-0.2229319248584839,0.0,1.0,0.0,1.0,0.0,0.0,0.0,0.0
-1.2687311813353392,-1.6992420007230695,0.672949549390788,0.0,0.0,1.0,0.0,0.0,0.0,1.0,0.0
-1.0349035916011813,-0.8920011452726803,-1.1041525189067318,0.0,0.0,0.0,0.0,0.0,0.0,0.0,1
0.2554782924873209,1.2361792918238006,-1.2656764360347754,0.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0
1.312032586841665,-0.671844548331665,1.1626059991105337,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0
-1.5718410198796182,-0.011374757508619205,-0.19212712744188168,1.0,0.0,0.0,0.0,0.0,0.0,0.
-1.0002624671961207,1.7498780180195028,-0.9283490739528356,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1.
0.6018895365379254,0.7958660979417701,0.25719071548198336,0.0,0.0,1.0,0.0,1.0,0.0,0.0,0.0
0.723133471955637,0.5023239686870831,1.6626860804182908,0.0,0.0,1.0,0.0,1.0,0.0,0.0,0.0
0.9569610616897951,-0.3049168867633062,-0.5389035539308398,0.0,1.0,0.0,0.0,0.0,1.0,0.0,0.
0.7837554396644929,-1.552470936095726,-0.21983873337098464,0.0,1.0,0.0,0.0,1.0,0.0,0.0,0.
-0.8616979695758789,-1.6992420007230695,-1.4152936845602533,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1
0.0043301405506325575,1.0894082271964571,[illegible]75348,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1
-1.4419368033606417,-0.378302419076978,-[illegible]188,1.0,0.0,0.0,0.0,0.0,0.0,0.0,1.0
-0.8010760018670231,0.42893843637341134,-0.06882319280321256,0.0,0.0,1.0,0.0,0.0,0.0,0.0,
-1.0695447160062417,0.8692516302554418,0.004608325660297258,0.0,0.0,1.0,0.0,0.0,0.0,0.0,1
0.19485632477846507,1.2361792918238006,0.9681162877732515,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0
0.012990421651897672,1.7498780180195028,-0.5313612513996772,0.0,1.0,0.0,0.0,0.0,0.0,1.0,0.0
-1.5545204576770881,0.06201077480505255,0.4390957984386179,1.0,0.0,0.0,1.0,0.0,0.0,0.0,0.
-0.9483007805885301,-0.5984590160179932,-0.40911662891426126,0.0,1.0,0.0,0.0,0.0,0.0,0.0,
-0.5585881310315999,-0.08476028982229096,-1.0162296101867188,0.0,0.0,1.0,0.0,1.0,0.0,0.0,
0.4027030712088278,1.6764924857058312,-0.8254686776288876,0.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0
1.5371998954745578,0.8692516302554418,-0.585428543701719,1.0,0.0,0.0,0.0,0.0,0.0,1.0,0.0
-0.35940166570250226,-1.1121577422136955,1.4111630027498578,0.0,1.0,0.0,0.0,1.0,0.0,0.0,0
-0.18619604367719997,0.42893843637341134,1.3576889252536377,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
-1.1388269648163625,0.8692516302554418,0.6000688732469698,1.0,0.0,0.0,0.0,0.0,1.0,0.0,0.0
0.09959323266454882,-1.4790854037820542,-1.5530466369692957,0.0,1.0,0.0,0.0,0.0,0.0,0.0,1
```