به نام خدا

درس: داده‌کاوی **استاد: دکتر رضا رمضانی**

پروژه اول: تحلیل و پیش‌پردازش **نام و نام خانوادگی: فاطمه مومنی**

این پروژه با زبان برنامه‌نویسی پایتون و با استفاده از کتابخانه‌های sklearn، pandas، numpy، matplotlib و seaborn پیاده‌سازی شده‌است. در ادامه کدهای مربوط به هر بخش، توضیح داده شده‌اند.

دستورات نشان داده‌شده در شکل ۱، برای نمایش تمام ستون‌های مجموعه داده به‌طور کامل، و دستورات نشان داده‌شده در شکل ۲، برای خواندن فایل‌های مجموعه داده، استفاده شده‌است.

```
pd.set_option('display.width', 2000)
pd.set_option('display.max_columns', 61)
pd.set_option('display.max_rows', 200)
```

شکل ۱

```
questions = pd.read_csv('/content/DM1_dataset/Questions.csv')
answers = pd.read_csv('/content/DM1_dataset/Answers.csv')
```

شکل ۲

اولین دستور شکل ۳، تعداد مقادیر جاافتاده هر ویژگی را می‌شمارد. تابع ()isna این مقادیر را یافته و تابع ()sum مجموع تعداد آن‌ها را به‌دست می‌دهد. سپس درصد مقادیر جاافتاده را محاسبه کرده و در دیکشنری NaN_percent که کلیدهای آن نام ویژگی‌ها است، ذخیره می‌کنیم.

```
#counting number of missing values per column
NaN_num = answers.isna().sum()

#define a dictionary for storing percentage of missing values in each column
NaN_percent = dict()
answers_columns = answers.columns
total_num = len(answers)

#calculating percentage of missing values per column
for num in range(len(NaN_num)):
    NaN_percent[answers_columns[num]] = NaN_num[num] * 100 / total_num
```

شکل ۳

در برخورد با داده‌های جاافتاده، سه روش به‌کار گرفته‌شده‌است: حذف ویژگی، پر کردن با میانگین مقادیر ویژگی و پر کردن با مد. مطابق شکل ۴، ابتدا دیکشنری حاوی درصد مقادیر جاافتاده را به‌صورت نزولی مرتب می‌کنیم. سپس ستون‌هایی که درصد مقادیر جاافتاده در آن‌ها نسبتا زیاد و بیش از ٪۴۰ است را در لیست max_miss ذخیره می‌کنیم. در نهایت، ستون‌های مشخص‌شده را با استفاده از تابع ()drop حذف می‌کنیم.

```
#sorting list of percentages to find columns with max number of missing values that stored in max_miss list
sort = sorted(NaN_percent.items(), key=lambda x: x[1], reverse=True)
max_miss = list()
for i in sort:
    if i[1] >= 40.0:
        max_miss.append(i[0])
    else:
        break
```

```
# deleting columns have many missing values
processed_ans = answers.drop(axis=1, columns=max_miss)
```

شکل ۴

پس از یافتن نوع داده هر ویژگی توسط تابع dtypes، مقادیر جاافتاده ستون‌هایی که از نوع عددی و دسته‌بندی‌شده هستند را به‌ترتیب با میانگین و مد سایر مقادیر ستون‌ها، پر می‌کنیم. شکل ۵ کدهای مربوطه را نشان می‌دهد.

```
# finding type of columns
print(processed_ans.dtypes)

# filling missing values with mean for numeric(float64) columns
numeric = ['Age', 'WorkWeekHrs']
for i in numeric:
    processed_ans[i] = processed_ans[i].fillna(processed_ans[i].mean())

# filling missing values with mode for categorical columns
categorical = ['MainBranch', 'Hobbyist', 'Age1stCode', 'CompFreq', 'Country', 'CurrencyDesc', 'CurrencySymbol',
               'EdLevel', 'Employment', 'Ethnicity', 'Gender', 'JobSat', 'JobSeek', 'NEWDevOps', 'NEWDevOpsImpt',
               'NEWEdImpt', 'NEWLearn', 'NEWOffTopic', 'NEWOnboardGood', 'NEWOtherComms', 'NEWOvertime', 'NEWPurpleLink',
               'OpSys', 'OrgSize', 'PurchaseWhat', 'Sexuality', 'SOAccount', 'SOComm', 'SOPartFreq', 'SOVisitFreq',
               'SurveyEase', 'SurveyLength', 'Trans', 'UndergradMajor', 'WelcomeChange', 'YearsCode', 'YearsCodePro']
for i in categorical:
    processed_ans[i] = processed_ans[i].fillna(processed_ans[i].mode()[0])
```

شکل ۵

همان‌گونه که در شکل ۶ مشاهده می‌شود، پس از یافتن ستون‌های باقی‌مانده، ستون‌های چند مقداری را برحسب کاراکتر ; جدا کرده و مقدار دارای بیشترین تکرار را برای هر ستون می‌یابیم (به‌دلیل عدم وجود مقادیر جاافتاده در اولین ستون مجموعه داده، یعنی ستون Respondent، حلقه for از عدد یک شروع شده‌است). سپس مقادیر جاافتاده هر ستون را با مقدار به‌دست آمده برای آن ستون، پر می‌کنیم. مقادیر متفاوت ستون LanguageWorkedWith (بدون احتساب تکرار هر مقدار) در لیست LanguageWorkedWith برای استفاده در بخش‌های بعدی، ذخیره شده‌است.

```
# remained columns
remained = list()
for i in answers_columns:
    if i not in (max_miss + numeric + categorical):
        remained.append(i)

multi_value = dict()
for i in range(1, len(remained)):
    multi_value.clear()
    column = processed_ans[remained[i]]
    for row in column:
        values = str(row).split(';')
        for val in values:
            if val != 'nan':
                if val not in multi_value.keys():
                    multi_value[val] = 1
                else:
                    multi_value[val] += 1
    if remained[i] == 'LanguageWorkedWith':
        LanguageWorkedWith = multi_value.keys()
    fill_value = max(multi_value, key=multi_value.get)
    processed_ans[remained[i]] = processed_ans[remained[i]].fillna(fill_value)
```

شکل ۶

برای بررسی وجود یا عدم وجود داده‌های پرت، نمودار جعبه‌ای ویژگی‌ها را رسم می‌کنیم. به‌این منظور، ابتدا باید مقادیر ستون‌های غیرعددی را به مقادیر عددی تبدیل کنیم که این کار با به‌کارگیری روش Label Encoding و توسط تابع fit_transform() انجام شده‌است. **شکل ۷** و شکل ۸ به‌ترتیب دستورات کدگذاری مقادیر ستون‌ها و ترسیم و نمایش نمودار جعبه‌ای را نشان می‌دهند.

```
# use label encoder to encode object data types
encod_ans = processed_ans.copy()
object_type = categorical + remained
label_encoder = preprocessing.LabelEncoder()
for i in object_type:
    encod_ans[i] = label_encoder.fit_transform(encod_ans[i])
```

شکل ۷

```
# show boxplot of all columns
for i in encod_ans.columns:
    plt.boxplot(encod_ans[i])
    plt.title(i)
    plt.show()
```

شکل ۸

با بررسی نمودارهای جعبه‌ای، ویژگی‌های بدون مقادیر جاافتاده به‌دست می‌آیند که نام آن‌ها در شکل ۹ مشاهده می‌شود. دستورات نشان داده‌شده در شکل ۱۰ نیز سایر ویژگی‌ها که داده‌های پرت دارند را به‌دست می‌آورد.

```
# features without outlier
no_outlier_columns = ['Respondent','CompFreq','Country','CurrencyDesc','CurrencySymbol','JobFactors',
                      'JobSat','JobSeek','LanguageDesireNextYear','LanguageWorkedWith','MiscTechWorkedWith',
                      'NEWCollabToolsWorkedWith','NEWDevOps','NEWEdImpt','NEWJobHuntResearch','NEWLearn',
                      'NEWOffTopic','NEWOnboardGood','NEWOtherComms','NEWOvertime','NEWPurpleLink',
                      'NEWStuck','OrgSize','PlatformDesireNextYear','PlatformWorkedWith','SOComm',
                      'SOVisitFreq','SurveyEase','YearsCode', 'YearsCodePro']

# discrete columns (no outlier)
discrete_columns = ['Hobbyist', 'Gender', 'PurchaseWhat', 'Sexuality', 'SOAccount', 'SurveyLength', 'Trans',
                    'UndergradMajor', 'WelcomeChange']
```

شکل ۹

```
# features with outlier
outlier_columns = list()
for i in encod_ans.columns:
    if i not in (no_outlier_columns + discrete_columns):
        outlier_columns.append(i)
```

شکل ۱۰

به‌منظور برخورد صحیح با داده‌های پرت هر ویژگی، توسط دستورات شکل ۱۱ نمودار scatter مقادیر هر یک را رسم می‌کنیم. بررسی نمودارها نشان می‌دهد که مقادیر بیشتر ستون‌ها از الگوی خاصی تبعیت نمی‌کنند و از این‌رو نمی‌توان گفت که داده‌های آن‌ها رفتار غیرطبیعی دارند. تنها داده‌ای در ستون Age نادرست (مقداری بیشتر از ۲۵۰) است که طبق دستورات شکل ۱۲ آن را حذف می‌کنیم. داده‌های پرت ستون‌های دیگر را با استفاده از روش IQR با میانه مقادیر هر ستون جایگزین می‌کنیم.

```
# detecting outliers
for i in outlier_columns:
    plt.plot(encod_ans[i], 'o')
    plt.title(i)
    plt.show()
```

شکل ۱۱

```
# outlier removal
for row in range(len(encod_ans['Age'])):
    if encod_ans.loc[row, 'Age'] > 250.0:
        remove = encod_ans.loc[row, 'Respondent']
encod_ans = encod_ans.set_index('Respondent').drop(remove)
```

شکل ۱۲