Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Систем штучного інтелекту»

Лабораторна робота №1

з дисципліни «Машинне навчання»

на тему «Основи роботи з PyTorch»

Виконав:
студент групи КН-308
Келемен С.Й.
Перевірила:
Якимишин Х. М.

Львів – 2019

Тема: основи роботи з PyTorch.

Мета: засвоїти основні відомості про роботу з фреймворком PyTorch, навчитись розпізнавати не типові елементи в даних.

Варіант 8

8	diamonds	'carat', 'cut',, 'clarity',

Завдання

Порахувати outliers для датасету і колонок згідно варіантів, а саме:

1. Порахувати z-score незалежно для кожної з згаданих цифрових колонок. Для обрахунку використати лише PyTorch.
2. Агрегувати пораховані z-score(наприклад усередненням). Візуалізувати датасет і знайдені нетипові дані(ті, для яких z-score > 3) на scatter plot. Нетипові дані і решту датасету звізуалізувати різними кольорами.
3. Якщо серед згаданих колонок є категоріальні дані, порахувати z-score для кожної з колонок в межах кожної з можливих категорій. Наприклад для iris датасету колонка ‘species’ може набувати значень 'setosa', 'versicolor', 'virginica'. Потрібно порахувати z-score для решти колонок лише для значень які в колонці ‘species’ мають значення ‘setosa’, після цього повторити те саме для значень які мають ‘versicolor’ і завершити обрахунком z-score для значень що мають ‘virginica’ в полі ‘species’
4. Аналогічно до пункту 2, агрегувати отримані значення і визначити нетипові дані.

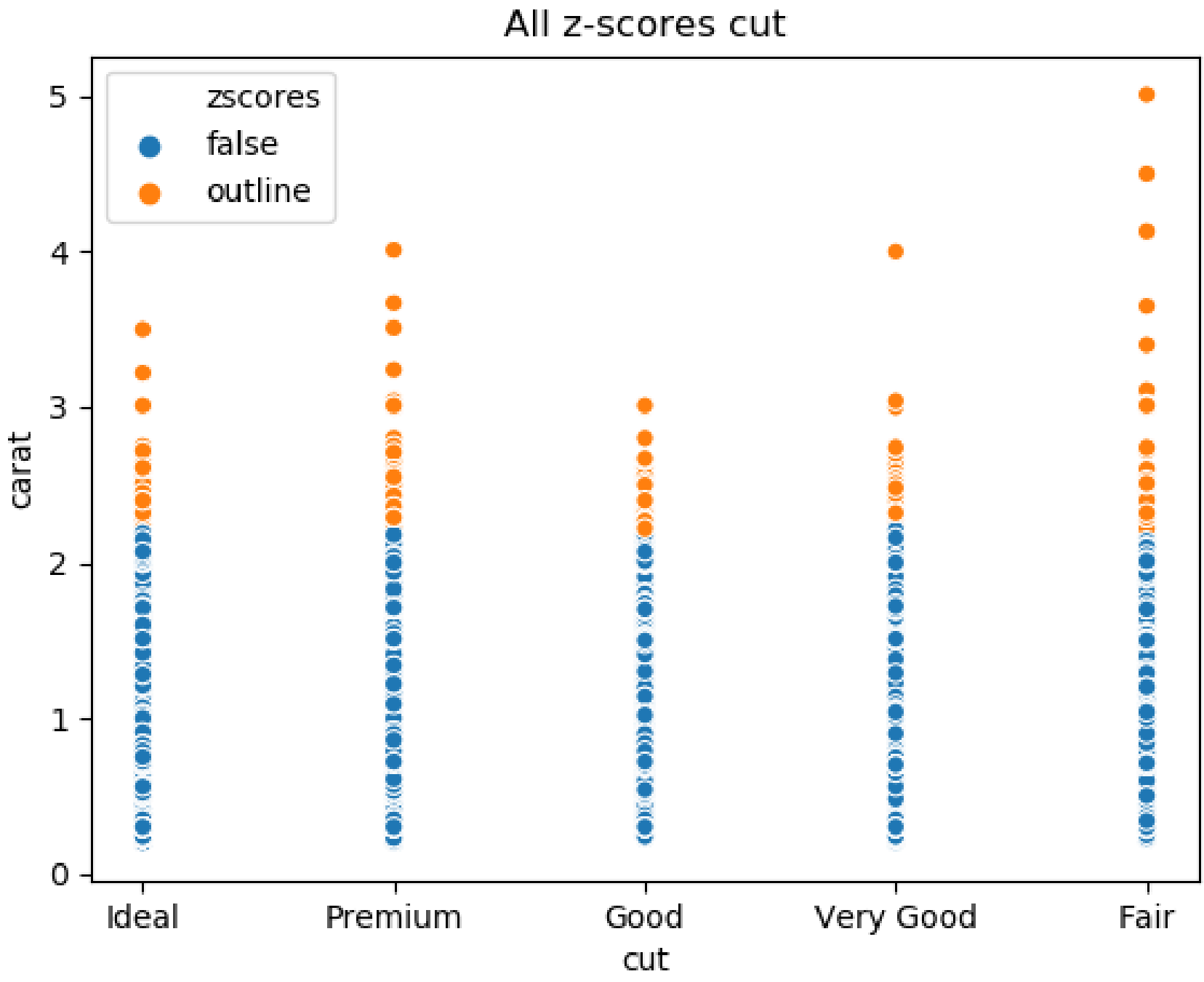
Код

```
import torch
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


initialDiamonds = pd.DataFrame(sns.load_dataset('diamonds'))
print('\nBase dataset:\n')
print(initialDiamonds)
diamonds = initialDiamonds[['carat', 'cut', 'clarity']]
print('\n\nVARIANT 8\nUsed dataset: diamonds(carat, cut, clarity)\n')
print('\nInitial dataset:\n')
print(diamonds)
diamondsCarat = torch.tensor(diamonds.carat)


class ZscoreCategory:

    def __init__(self, dataframe, numericCol, categoryCol):
        self.carat = dataframe[numericCol]
        self.categoryData = dataframe[categoryCol]
        self.mean = self.carat.mean()
        self.sigma = self.carat.std()

    def getScoreCat(self):
        return (self.carat - self.mean) / self.sigma


class Zscore:

    def __init__(self, columns: torch.tensor):
        self.carat = columns
        self.mean = torch.mean(self.carat)
        self.sigma = torch.std(self.carat)

    def getScore(self):
        return (self.carat - self.mean) / self.sigma

    def getAverageScore(self):
        return torch.mean((self.carat - self.mean) / self.sigma)


def checkOutliers(zscorestoCheck):
    return zscorestoCheck['zscores'].apply(lambda x: 'outline' if abs(x) > 3
else 'inlier')


categoryList = ['cut', 'clarity']
# 'Cut' unique values: ['Ideal', 'Premium', 'Good', 'Very Good', 'Fair']
# 'Clarity' unique values: ['SI2', 'SI1', 'VS1', 'VS2', 'VVS2', 'VVS1', 'I1',
'IF']
```

```
scoreCalcCarat = Zscore(diamondsCarat)

print('\n\nAll carat information :')
print('Mean: {0}\nsd: {1}'.format(scoreCalcCarat.mean, scoreCalcCarat.sigma))
print('Z-scores mean: {0}'.format(scoreCalcCarat.getAverageScore()))
zscores = scoreCalcCarat.getScore()
print('Z-scores: {0}'.format(zscores))

for j in categoryList:

    initialDataframe = pd.DataFrame({'carat': scoreCalcCarat.carat,
'zscores': zscores, j: diamonds[j]})

    sns.scatterplot(x=j, y='carat', hue=checkOutliers(initialDataframe),
data=initialDataframe)
    plt.title('All z-scores {0}'.format(j))
    plt.xlabel(j)
    plt.ylabel('carat')
    plt.show()

    categoryObj = ZscoreCategory(diamonds, 'carat', j)
    emptyDataframe = pd.DataFrame(columns=['carat', 'zscores', j])
    sourceDataframe = pd.DataFrame({'carat': categoryObj.carat, j:
categoryObj.categoryData})

    for i in diamonds[j].unique().tolist():
        tempDataframe = ZscoreCategory(sourceDataframe.loc[sourceDataframe[j]
== i], 'carat', j)
        zscoresCat = pd.DataFrame({'carat': tempDataframe.carat, 'zscores':
tempDataframe.getScoreCat(), j: i})
        emptyDataframe = emptyDataframe.append(zscoresCat, ignore_index=True)

    sns.scatterplot(x=j, y='carat', hue=checkOutliers(emptyDataframe),
data=emptyDataframe)
    plt.title('Category z-scores {0}'.format(j))
    plt.xlabel(j)
    plt.ylabel('carat')
    plt.show()
```

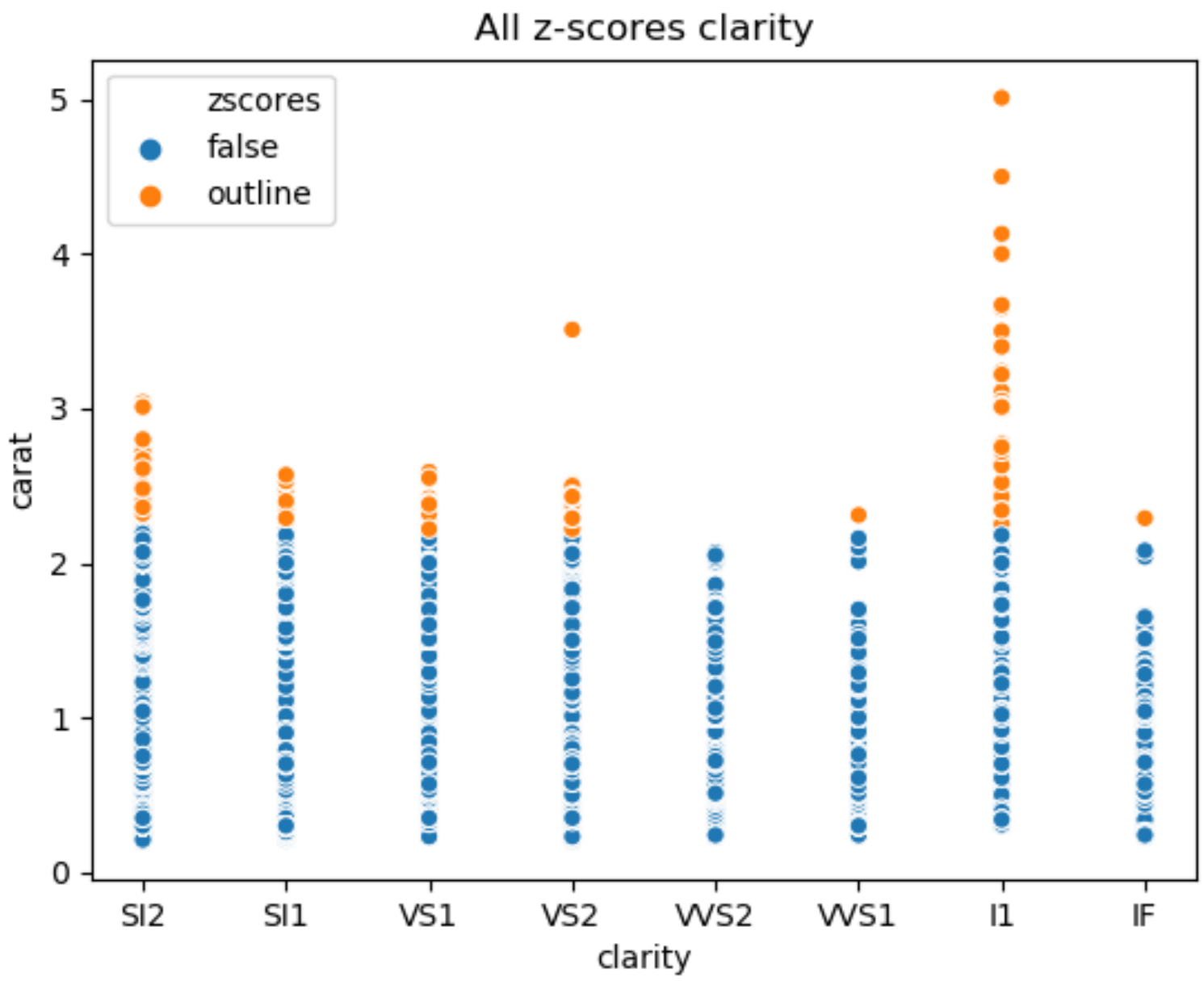
Результати роботи програми

Z-scores визначені для всіх значень carat разом, погруповані по категоріям cut.



Z-scores визначені для значень carat у кожній з категорій cut окремо.

Z-scores визначені для всіх значень carat разом, погруповані по категоріям clarity.



Z-scores визначені для значень carat у кожній з категорій clarity окремо.

Висновки

У цій лабораторній роботі я засвоїв основні відомості про роботу з фреймворком PyTorch, навчився розпізнавати не типові елементи в даних. Було пораховано Z-scores для числової колонки carat спочатку для всіх її значень разом, а потім для кожної з категорій колонок cut та clarity окремо. На scatter plots можна чітко побачити різницю між розподілом нетипових значень у першому та другому випадках.