Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика с системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Технологии машинного обучения»

Отчёт по рубежному контролю №1
Вариант №13

Выполнил:
студент группы РТ5-61Б
Мицкевич В.Б.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2023 г.

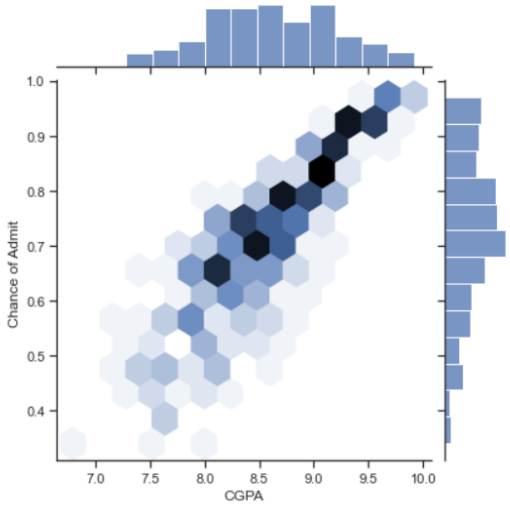
Вариант 13 (Задача №2, Датасет №4)

Для заданного набора данных проведите обработку пропусков в данных для одного категориального и одного количественного признака. Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали? Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

Для студентов группы РТ5-61Б - для пары произвольных колонок данных построить график "Jointplot".

Построим график joinplot на основе средних баллов ученика (Undergraduate GPA) и его вероятности поступления (Chance of Admit)

```
sns.jointplot(x = "CGPA", y = "Chance of Admit ",
              kind = "hex", data = data)
plt.show()
```



1) Обработка пропусков в данных для одного количественного признака

Для начала посмотрим существуют ли пропуска в данном датасете.

```
|: def count_nan(data):
    for col in data.columns:
        count_nan = data[data[col].isnull()].shape[0]
        print('{} имеет NAN: {}'.format(col, count_nan))
count_nan(data)
```

```
Serial No. имеет NAN: 0
GRE Score имеет NAN: 0
TOEFL Score имеет NAN: 0
University Rating имеет NAN: 0
SOP имеет NAN: 0
LOR  имеет NAN: 0
CGPA имеет NAN: 0
Research имеет NAN: 0
Chance of Admit  имеет NAN: 0
```

В данном датасете отсутствуют признаки. Поэтому искусственно их создадим.В качестве анализа возбмем столбец «LOR».

```
import random
def count_nan(length, count):
    return (count / length) * 100

def create_nan(data, column):
    length = data.shape[0]
    while (count_nan(length, data[f'{column}'].isnull().sum()) < 5):
        index = random.randint(0, 399)
        data_column = data[f'{column}']
        data_column[index] = None
```

```
create_nan(data, 'LOR ')
```

```
data.isnull().sum() / data.count()
```

```
Serial No.           0.000000
GRE Score            0.000000
TOEFL Score          0.000000
University Rating    0.000000
SOP                  0.000000
LOR                  0.052632
CGPA                 0.000000
Research             0.000000
Chance of Admit      0.000000
dtype: float64
```

Просмотрев описание данных, мы видим, что значения «LOR» находятся в интервале [1, 5], поэтому заполним пропуска средним значение — 2,5.

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

```
strateg=['mean', 'median', 'most_frequent']
def fill_nan(strategy_param, data):
    imputation = SimpleImputer(strategy=strategy_param)
    data_fill = imputation.fit_transform(data)
    return data_fill
```

```
result = fill_nan(strateg[0], data)
```

```
data_column = data['LOR ']
for i in range(0, 400):
    data_column[i] = result[i][5]
```

...

```
data.isnull().sum() / data.count()
```

```
Serial No.           0.0
GRE Score            0.0
TOEFL Score          0.0
University Rating    0.0
SOP                  0.0
LOR                  0.0
CGPA                 0.0
Research             0.0
Chance of Admit      0.0
dtype: float64
```

2)Обработка пропусков в данных для одного категориального признака

В данном датасете отсутсвуют категориальные признаки, поэтому искуственно создадим один.

```
from sklearn.impute import SimpleImputer
from sklearn.impute import MissingIndicator
```

```
strateg=['mean', 'median', 'most_frequent']
def fill_nan(strategy_param, data):
    imputation = SimpleImputer(strategy=strategy_param)
    data_fill = imputation.fit_transform(data)
    return data_fill
```

```
result = fill_nan(strateg[0], data)
```

```
data_column = data['LOR ']
for i in range(0, 400):
    data_column[i] = result[i][5]
```

...

```
data.isnull().sum() / data.count()
```

```
Serial No.           0.0
GRE Score            0.0
TOEFL Score          0.0
University Rating    0.0
SOP                  0.0
LOR                  0.0
CGPA                 0.0
Research             0.0
Chance of Admit      0.0
dtype: float64
```

Основываясь на баллах бакалвриата можно предположить сделал ученик исследование или нет. На своем опыте могу сказать,что ученики с малым баллом не занимаются исследованиями. Поэтому если у ученика ср балл < 5, то он не занимался исследованием.

```
ball = 5
column_res = data.Research
column_CGPA = data.CGPA
for i in range(0, 399):
    if (column_CGPA[i] <= 5 and column_res[i] == None):
        column_res[i] = 'No'
    else:
        column_res[i] = 'Yes'
```

```
data.isnull().sum() / data.count()
```

```
Serial No.           0.0
GRE Score            0.0
TOEFL Score          0.0
University Rating    0.0
SOP                  0.0
LOR                  0.0
CGPA                 0.0
Research             0.0
Chance of Admit      0.0
dtype: float64
```

Вывод:

Для заполнения пропусков можно использовать разные виды заполнения: среднее значение, наиболее встречаемое значение…

А также можно выполнять заполнения пропусков, основываясь на своем опыте.

В моем случае для заполнения пропусков количественного признака я заполнял средним значением, а при заполнение категориального признака решил использовать свой собственный опыт. Основываясь на баллах, можно понять делал ли студент исследования или нет.