Рубежный контроль №1 по предмету "Технологии машинного обучения"

Имя студента: Пересыпко Александр, РТ5-61Б
Датасет №5: Graduate Admission
Вариант 13
Задача 2
Для студентов группы РТ5-61Б для пары произвольных колонок данных построить график "Jointplot".

Описание датасета

Датасет содержит несколько параметров, которые считаются важными при подаче заявления на программы магистратуры, и вероятность поступить в магистратуру

Основные атрибуты:

1. **GRE Scores (из 340)** — баллы GRE (из 340)
2. **TOEFL Scores (из 120)** — баллы TOEFL (из 120)
3. **University Rating (из 5)** — рейтинг университета (из 5)
4. **Statement of Purpose and Letter of Recommendation Strength (из 5)** — сила мотивационного письма и рекомендаций (из 5)
5. **Undergraduate GPA (из 10)** — средний балл за бакалавриат (из 10)
6. **Research Experience (0 или 1)** — опыт научной работы (0 или 1)
7. **Chance of Admit (от 0 до 1)** — вероятность поступления (от 0 до 1)

Вариант задания №13

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

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Загрузка данных из CSV файла
df = pd.read_csv('Admission_Predict.csv')

# Отображение первых нескольких строк исходного DataFrame
print("Исходный DataFrame:")
print(df.head(10))
```

```
Исходный DataFrame:
   Serial No.  GRE Score  TOEFL Score  University Rating  SOP  LOR
CGPA  \
0           1        337          118                  4  4.5   4.5
9.65
1           2        324          107                  4  4.0   4.5
8.87
2           3        316          104                  3  3.0   3.5
8.00
3           4        322          110                  3  3.5   2.5
8.67
4           5        314          103                  2  2.0   3.0
8.21
5           6        330          115                  5  4.5   3.0
9.34
6           7        321          109                  3  3.0   4.0
8.20
7           8        308          101                  2  3.0   4.0
7.90
8           9        302          102                  1  2.0   1.5
8.00
9          10        323          108                  3  3.5   3.0
8.60

   Research  Chance of Admit
0         1              0.92
1         1              0.76
2         1              0.72
3         1              0.80
4         0              0.65
5         1              0.90
6         1              0.75
7         0              0.68
8         0              0.50
9         0              0.45
```

Подсчитаем количество строк с пропусками в датасете

```
initial_row_count = df.shape[0]

total_count = df.shape[0]
print('Всего строк: {}'.format(total_count))

df.isnull().sum()
```

```
Всего строк: 400

Serial No.            0
GRE Score             0
TOEFL Score           0
University Rating     0
SOP                   0
LOR                   0
CGPA                  0
Research              0
Chance of Admit       0
dtype: int64
```

Добавим категориальный признак

Для этого разобьем GRE score на 3 категории: низкий средний и высокий

```
bins = [ -np.inf, 50, 75, np.inf ]
labels = ['низкий', 'средний', 'высокий']
df['score_cat'] = pd.cut(df['GRE Score'], bins=bins, labels=labels,
right=False)
```

Строк с пропусками нет. Добавим их случайно

Добавим пропуски в категориальный признак GRE Score и в числовой CGPA

```
# задаём долю пропусков для каждого признака
frac_gre = 0.1    # 10% GRE Score
frac_cgpa = 0.05  # 5% CGPA

# выбираем случайные индексы для каждого
gre_idx = df.sample(frac=frac_gre, random_state=42).index
cgpa_idx = df.sample(frac=frac_cgpa, random_state=24).index

# вставляем NaN
df.loc[gre_idx, 'GRE Score'] = np.nan
df.loc[cgpa_idx, 'CGPA'] = np.nan

print('Пропущенные в GRE Score и CGPA:')
print(df[['GRE Score', 'CGPA']].isnull().sum())
```

```
Пропущенные в GRE Score и CGPA:
GRE Score    40
CGPA         20
dtype: int64
```

Обработка пропусков

```
from sklearn.impute import SimpleImputer
imp = SimpleImputer(strategy='most_frequent')

df['GRE Score'] = imp.fit_transform(df[['GRE Score']])

print('Пропущенных в GRE Score после заполнения:', df['GRE
Score'].isnull().sum())
```

```
Пропущенных в GRE Score после заполнения: 0
```

```
imp_cgpa = SimpleImputer(strategy='mean')
df['CGPA'] = imp_cgpa.fit_transform(df[['CGPA']])
print('Пропущенных в CGPA после заполнения:',
df['CGPA'].isnull().sum())
```

```
Пропущенных в CGPA после заполнения: 0
```

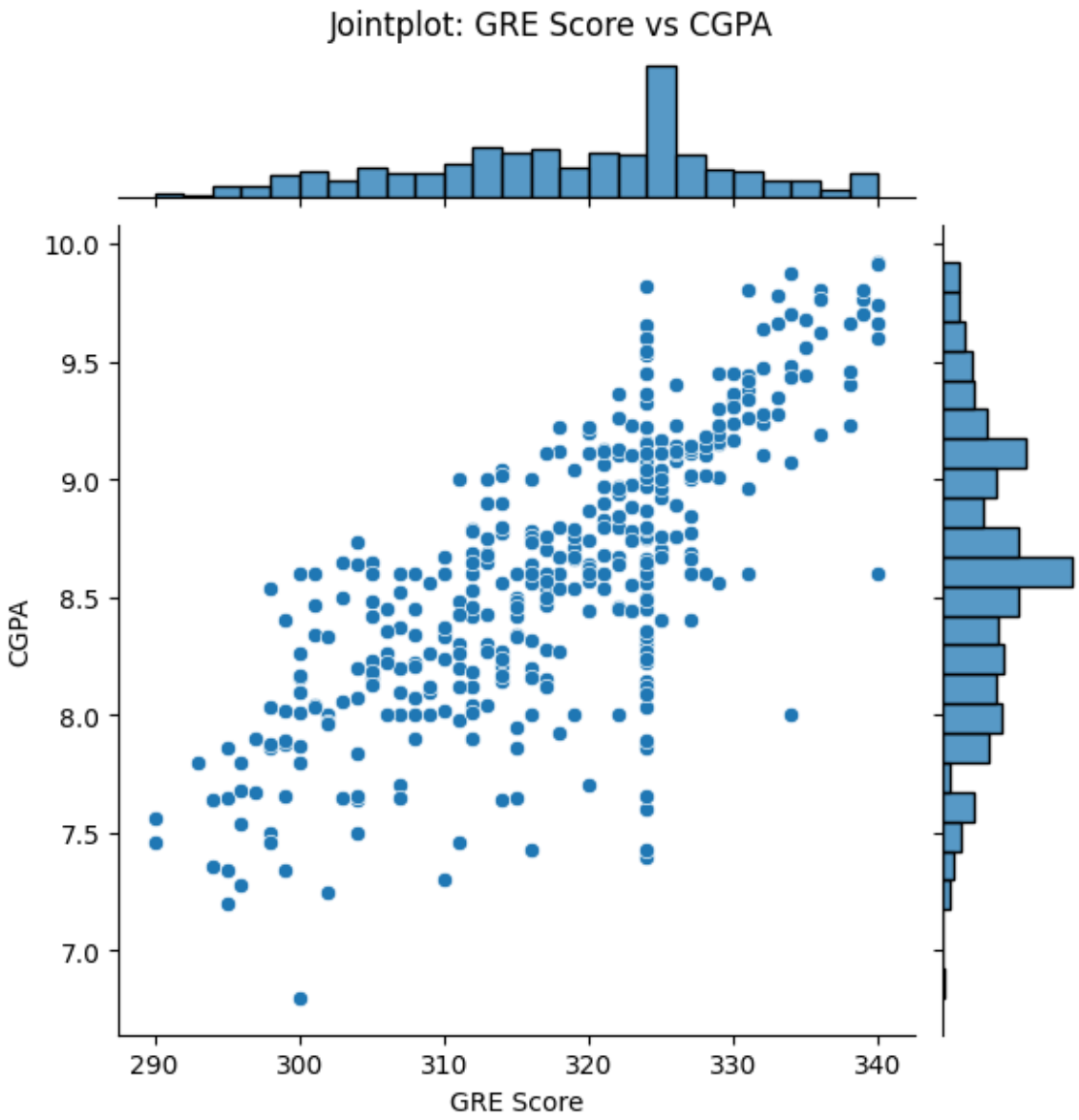
Для студентов группы РТ5-61Б - для пары произвольных колонок данных построить график "Jointplot".

```
import seaborn as sns
import matplotlib.pyplot as plt

col_x = 'GRE Score'
col_y = 'CGPA'

sns.jointplot(
    x=col_x,
    y=col_y,
    data=df,
    kind='scatter',
    height=6,
    marginal_kws=dict(bins=25, fill=True)
)

plt.suptitle(f'Jointplot: {col_x} vs {col_y}', y=1.02)
plt.show()
```



Ответы на вопросы

Какие способы обработки пропусков в данных для категориальных и количественных признаков Вы использовали?

• Категориальный признак (GRE Score) – SimpleImputer(strategy='most_frequent') – Заполнили пустые значения наиболее часто встречающимся баллом GRE.

• Количественный признак (CGPA) – SimpleImputer(strategy='mean') – Заполнили пропуски средним значением CGPA по всей выборке.

Какие признаки Вы будете использовать для дальнейшего построения моделей машинного обучения и почему?

• GRE Score, CGPA – Ключевые академические метрики, сильная корреляция с целевой переменной (Admission Chance).

• TOEFL Score – Аналогичный по значимости тест, также числовой, легко нормировать.

• University Rating, SOP, LOR – Категориальные/ранжировочные факторы, отражают престиж университета и качество рекомендаций/мотивации. – Будут преобразованы в dummy-переменные или порядковые коды.

• Research (0/1) – Бинарный признак, важный индикатор опыта, не требует дополнительной обработки.

Отобранные признаки дают сбалансированный набор академических, рейтинговых и бинарных факторов, что обычно улучшает обобщающую способность моделей регрессии или классификации в задачах предсказания шансов поступления.