Московский Государственный Технический Университет имени

Н.Э.Баумана

Факультет Информатика и системы управления

Кафедра ИУ-5

«Системы обработки информации и управления»

ОТЧЁТ

Лабораторная работа №1

Медоты мошинного обучения

Выполнил: **Юй Шанчэнь**

стулент группы:ИУ-5 23М

Москва 2022г.

1. Цель лабораторной работы

Изучение различных методов визуализация данных и создание истории на основе данных.

2. Задание

2.1 Выбрать набор данных (датасет).

2.2 Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:

История должна содержать не менее 5 шагов (где 5 - рекомендуемое

количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.

На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.

Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.

Выбор графиков должен быть обоснован использованием методологии datato-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz.

Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.

История должна содержать итоговые выводы. В реальных "историях о

данных" именно эти выводы представляют собой основную ценность для предприятия.

2.3 Сформировать отчет и разместить его в своем репозитории на github.

3. Ход выполнения работы Контекст

Чтобы узнать, какой фактор может повлиять на успеваемость учащегося, мы классифицируем оценку по нескольким рангам и выясняем, какая характеристика влияет на оценку более значимо. Независимые переменные следующие:

1. gender : sex of students
2. race/ethnicity : ethnicity of students
3. parental level of education : parents' final education
4. lunch : having lunch before test (normal or abnormal)
5. test preparation course : complete or not complete before test

3.2. Основные характеристики набора данных

Импортируйте соответствующую библиотеку и передайте данные

```
[13]: import seaborn as sns
      import matplotlib.pyplot as plt
      import os
      import pandas as pd
      import numpy as np
      import seaborn as sns
      from sklearn.preprocessing import LabelEncoder
      from sklearn.cluster import KMeans
      score_df =pd.read_csv("D:/lab1/StudentsPerformance.csv")
```

```
[4]: score_df.head()
```

Отображение данных с помощью команды .head()

	gender	race/ethnicity	parental level of education	lunch	test preparation course	math score	reading score	writing score
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Некоторые имена столбцов слишком длинные, переименуйте их, чтобы они были проще.

```
: score_df.rename(columns={"race/ethnicity":"ethnicity","parental level of education":"parent_education"
                           ,"math score":"math","reading score":"reading","writing score":"writing",
                           "test preparation course":"pre"},inplace=True)
  score_df.head()
```

:

	gender	ethnicity	parent_education	lunch	pre	math	reading	writing
0	female	group B	bachelor's degree	standard	none	72	72	74
1	female	group C	some college	standard	completed	69	90	88
2	female	group B	master's degree	standard	none	90	95	93
3	male	group A	associate's degree	free/reduced	none	47	57	44
4	male	group C	some college	standard	none	76	78	75

Кажется ясно. Проверьте тип данных.

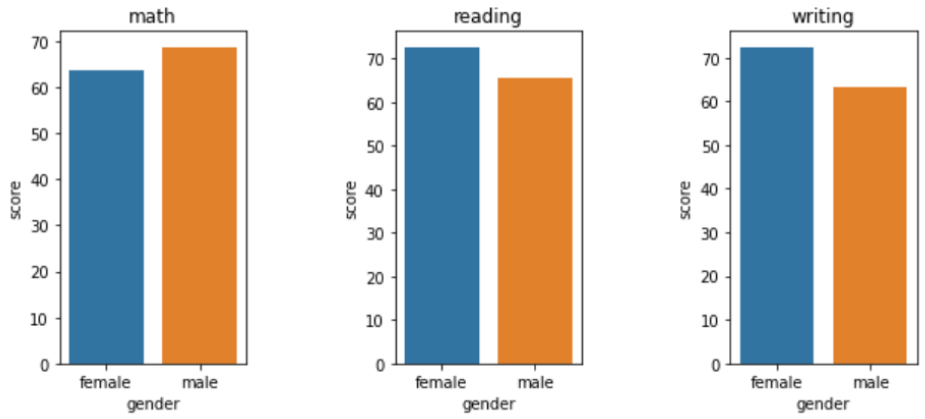
```
[6]: score_df.dtypes
```

```
[6]: gender              object
     ethnicity           object
     parent_education    object
     lunch               object
     pre                 object
     math                 int64
     reading              int64
     writing              int64
     dtype: object
```

С типом данных мы разберемся позже. Во-первых, давайте выясним производительность каждого поля для мужчин и женщин.

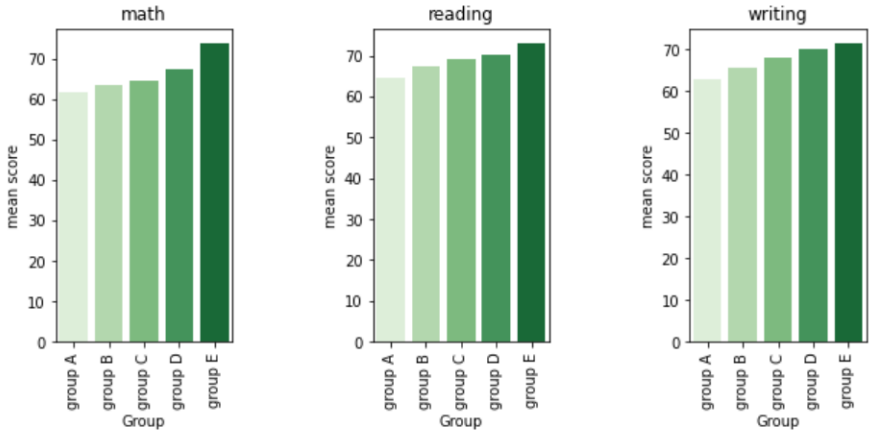
```
fig, ax = plt.subplots()
fig.subplots_adjust(hspace=0.8, wspace=0.8, left = 0.2, right = 1.5)
for idx in range(3):
    plt.subplot(1,3, idx+1)
    gender_df = score_df.groupby("gender")[list(score_df.columns[-3:])[idx]].describe()
    sns.barplot(gender_df.index, gender_df.loc[:,"mean"].values)
    plt.ylabel("score")
    plt.title(list(score_df.columns[-3:])[idx])

plt.show()
```



Мы видим, что у мужчин лучше результаты в математике, но хуже в чтении и письме. Во-вторых, увидеть производительность этнической принадлежности.

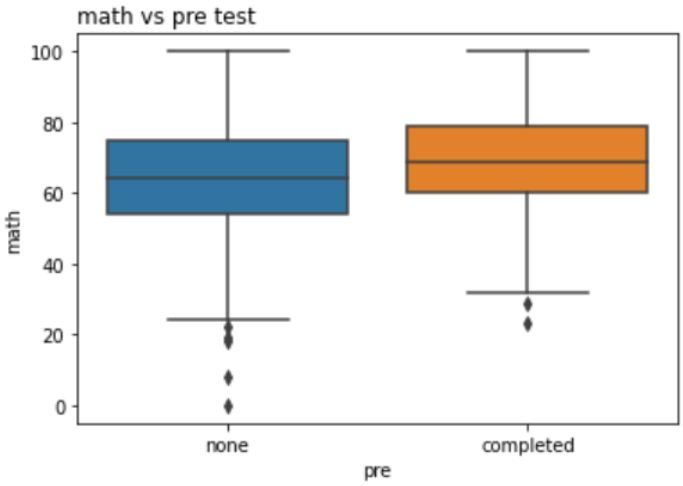
```
fig, ax = plt.subplots()
fig.subplots_adjust(hspace=0.8, wspace=0.8, left = 0.2, right = 1.5)
for idx in range(3):
    plt.subplot(1,3, idx+1)
    ethn_df = score_df.groupby("ethnicity")[list(score_df.columns[-3:])[idx]].mean()
    sns.barplot(x=ethn_df.index, y = ethn_df.values, palette = "Greens")
    plt.xlabel("Group")
    plt.ylabel("mean score")
    plt.xticks(rotation=90)
    plt.title(list(score_df.columns[-3:])[idx])
plt.show()
```

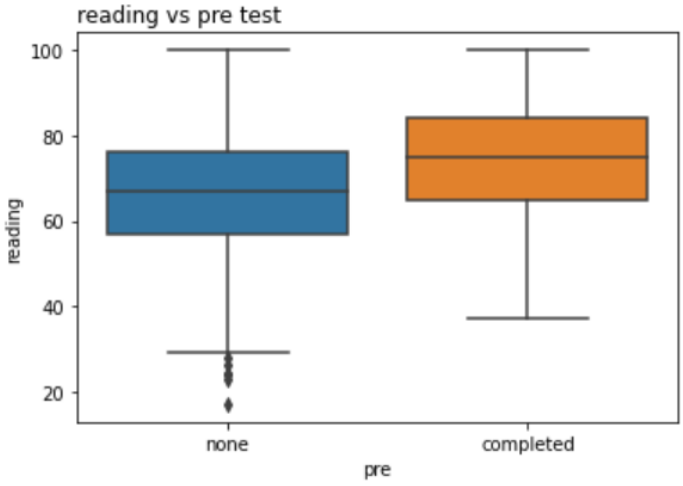


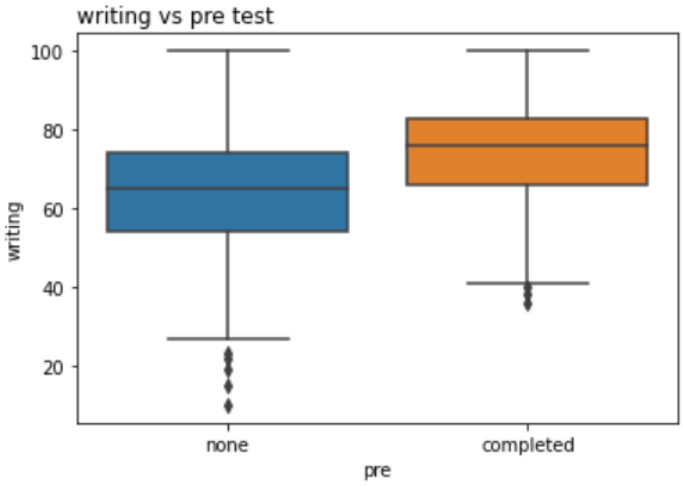
Очевидно, что группа E имеет лучшие показатели по всем полям, а группа A — худшие.

Затем давайте посмотрим на результат оценки и подготовки к тесту.

```
for item in score_df.columns[-3:]:
    sns.boxplot(x=score_df["pre"], y=score_df[item])
    plt.title(item+" vs pre test", loc="left")
    plt.show()
```

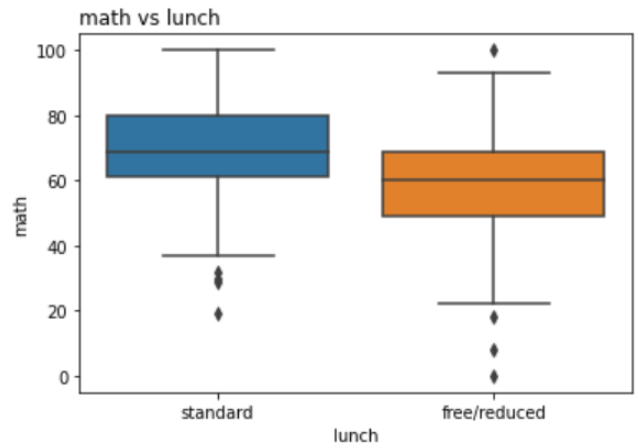


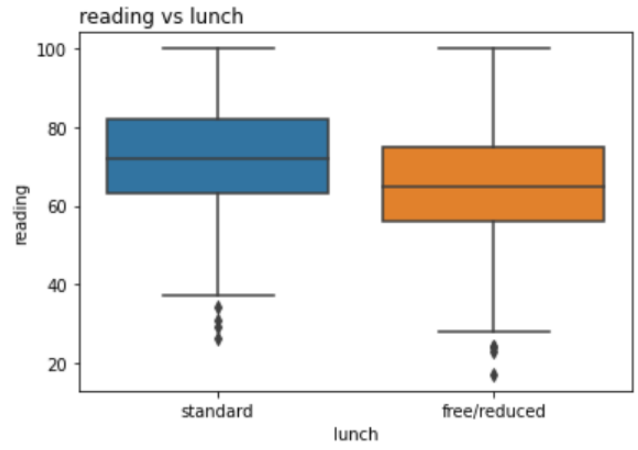


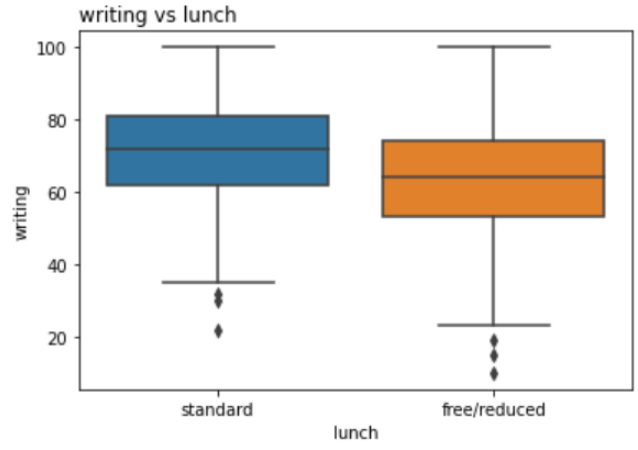


Распределение баллов становится более узким, если студенты завершают подготовку перед тестом, а также мы видим, что средний балл лучше.

```
for item in score_df.columns[-3:]:
    sns.boxplot(x=score_df["lunch"], y=score_df[item])
    plt.title(item+" vs lunch", loc="left")
    plt.show()
```







Имейте смысл! Учащимся легче получить более высокий балл, если они питаются стандартно.

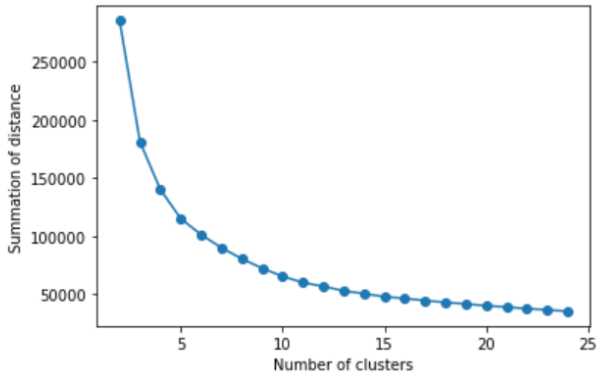
Мы заранее проверяем тип данных. Затем мы преобразуем некоторые функции с помощью кодировщика меток.

```
labelencoder = LabelEncoder()
train_df = score_df.copy()
train_df["parent_education"] = labelencoder.fit_transform(train_df["parent_education"])
train_df["pre"] = labelencoder.fit_transform(train_df["pre"])
train_df["lunch"] = labelencoder.fit_transform(train_df["lunch"])
train_df.head()
```

	gender	ethnicity	parent_education	lunch	pre	math	reading	writing
0	female	group B	1	1	1	72	72	74
1	female	group C	4	1	0	69	90	88
2	female	group B	3	1	1	90	95	93
3	male	group A	0	0	1	47	57	44
4	male	group C	4	1	1	76	78	75

Здорово! Функции «обучение родителей», «обед» и «предварительно» обозначены цифрами. Затем мы используем алгоритм K-средних для классификации набора данных.

```
kmeans_dis = list()
for idx in range(2, 25):
    kmeans = KMeans(init = "k-means++", n_clusters = idx, n_init = 20)
    kmeans.fit_transform(train_df.iloc[:, 2:])
    kmeans_dis.append(kmeans.inertia_)
plt.plot(list(range(2,25)), kmeans_dis, marker = "o")
plt.xlabel("Number of clusters")
plt.ylabel("Summation of distance")
plt.show()
```



Хорошо! Мы выбираем 8 в качестве точки изгиба, а затем классифицируем все данные.

Чтобы более четко увидеть влияние между оценками по математике и оценками по чтению фактором, нам нужно вызвать sns.hetmap для связанный анализ

```
score_df.corr()
```

	math score	reading score	writing score
math score	1.000000	0.817580	0.802642
reading score	0.817580	1.000000	0.954598
writing score	0.802642	0.954598	1.000000

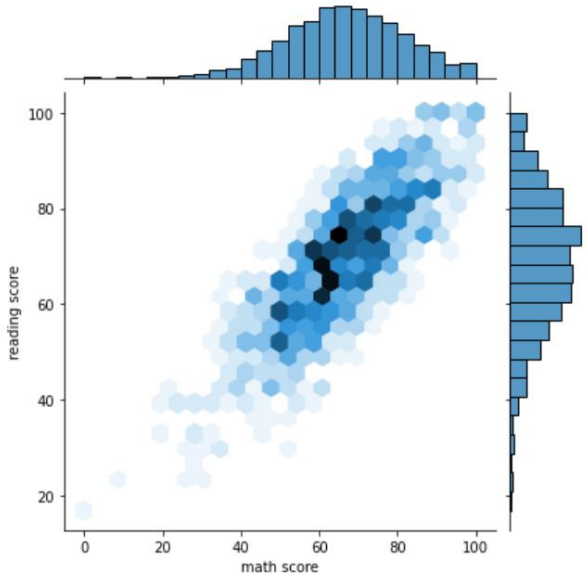
```
sns.heatmap(score_df.corr(),annot=True,fmt=".2f")
```

```
<AxesSubplot:>
```

Общие точки изменения отображаются на карте

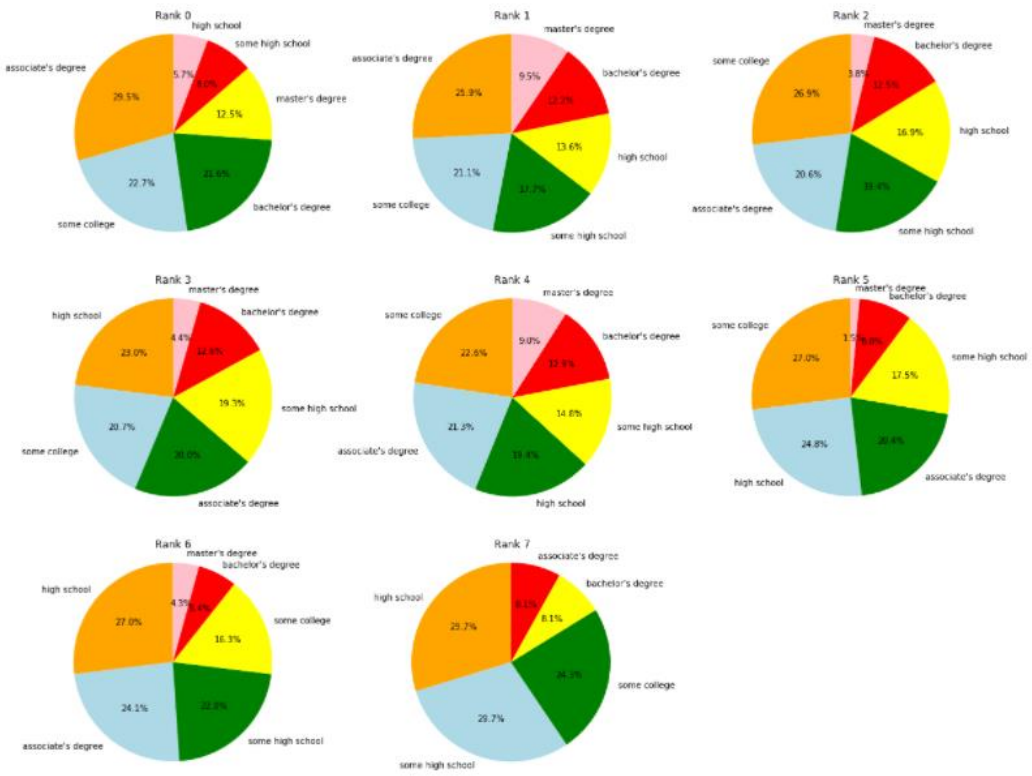
```
[21]: sns.jointplot(x="math score",y='reading score',data=score_df,kind = "hex")
```

```
[21]: <seaborn.axisgrid.JointGrid at 0x1a6b6bbe490>
```



```
def plot_pie_chart(column):
    fig, ax = plt.subplots(figsize=(20,16))
    color = ["orange","lightblue","green","yellow","red","pink","brown","gray"]
    for idx in range(8):
        plt.subplot(3, 3, idx+1)
        num = "class"+ str(idx)
        num = score_df[score_df["classification"]==rank.index[idx]]
        percentage_of_parent_edu = num[column].value_counts()
        percentage_of_parent_edu.sort_index()
        label = percentage_of_parent_edu.index
        value = percentage_of_parent_edu.values
        plt.pie(value, labels = label, autopct = "%1.1f%%",
                startangle=90, radius = 4, colors = color[:len(label)])
        plt.axis("equal")
        plt.title("Rank "+str(idx))
    plt.show()
plot_pie_chart("parent_education")
```

Отныне мы можем выяснить корреляцию между успеваемостью учеников и характеристиками.Давайте построим круговую диаграмму, чтобы увидеть, может ли уровень образования родителей влиять на успеваемость или нет.



Определим высшую степень образования. Родители, имеющие степень бакалавра или магистра, имеют высшее образование. Поэтому мы сосредоточимся на этих двух терминах.

Поскольку круговая диаграмма была показана выше, мы можем легко понять соотношение высшего образования. Для ранга 0 его соотношение составляет около 32%. Кроме того, между рангом 1 и рангом 3 нет различий, а соотношение составляет около 15–17%. Наконец, соотношение составляет всего 8% в ранге 7.

Мы рассчитали средний балл для каждого ранга ранее, поэтому мы можем сказать, что образование родителей влияет на балл, но не очевидно, потому что все еще есть 70% ~ 80% родителей без высшего образования.

Список литературы

[1] Гапанюк Ю. Е. Лабораторная работа «Разведочный анализ данных. Исследование

и визуализация данных» [Электронный ресурс] // GitHub. — 2019. — Режим доступа:

https://github.com/ugapanyuk/ml_course/wiki/LAB_EDA_VISUALIZATION (дата

обращения: 13.02.2019)

[2] https://www.kaggle.com/datasets