Министерство образования и науки Российской Федерации

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э. БАУМАНА

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления» (ИУ5)

ДИСЦИПЛИНА: «Технологии машинного обучения»

Отчет по лабораторной работе №1
«Разведочный анализ данных. Исследование и визуализация данных»

Выполнила:

Студентка группы ИУ5-61Б

Мартынова Д.П.

Преподаватель:

Гапанюк Ю.Е.

Москва, 2020 г.

Цель лабораторной работы: изучение различных методов визуализация данных.

Задание:

- Выбрать набор данных (датасет).
- Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных.
- Создать ноутбук, который содержит следующие разделы:

1. Текстовое описание выбранного Вами набора данных.
2. Основные характеристики датасета.
3. Визуальное исследование датасета.
4. Информация о корреляции признаков.

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

Выполнение ЛР:

1) Текстовое описание набора данных

Используемый **dataset: TED Talks** - Data about TED Talks on the TED.com website until September 21st, 2017.

Контекст

Данный набор данных содержит информацию обо всех аудио-видеозаписях выступлений TED Talks, загруженных на официальный сайт TED.com до 21 сентября 2017 года. Основной набор данных содержит информацию обо всех выступлениях, включая количество участников, количество комментариев, описание, спикеров и название.

Dataset состоит из следующих данных:

- Comments – Количество комментариев первого уровня, сделанных в ходе выступления
- Description – Описание того, о чем выступление
- Duration – Продолжительность выступления в секундах
- Event – Событие The TED/TEDx, когда состоялось выступление
- Film_date – Отметка времени начала съемок в формате Юникс
- Languages – Количество языков, на которых доступно выступление
- Main_speaker – Основной спикер выступления

- Name – Официальное название выступления: включает название и имя спикера
- Num_speaker – Количество спикеров в выступлении
- Published_date – Временная метка в формате Юникс публикации выступления на официальном сайте
- Ratings – строковый словарь различных рейтингов, присвоенных разговору
- Related_talks – Список выступлений, рекомендуемых к просмотру
- Speaker_occupation – Род занятия главного спикера
- Tags – Темы, связанные с выступлением
- Title – Название выступления
- url – Ссылка на выступление
- views – Количество просмотров выступления

2) *Основные характеристики датасета*

Загрузка данных и импорт библиотек:

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
```

In [3]:
```
data = pd.read_csv('ted_main.csv', sep=",")
```

In [4]:
```
# Первые 5 строк датасета
data.head()
```

Out[4]:

	comments	description	duration	event	film_date	languages	main_speaker	name	num_speaker	published_date	ratings	relate
0	4553	Sir Ken Robinson makes an entertaining and pro...	1164	TED2006	1140825600	60	Ken Robinson	Ken Robinson: Do schools kill creativity?	1	1151367060	[{'id': 7, 'name': 'Funny', 'count': 19645}, {...	[{'id': 865 'https://pe.tedcdn.c
1	265	With the same humor and humanity he exuded in ...	977	TED2006	1140825600	43	Al Gore	Al Gore: Averting the climate crisis	1	1151367060	[{'id': 7, 'name': 'Funny', 'count': 544}, {'i...	[{'id': 243 'https://pe.tedcdn.c
2	124	New York Times columnist David Pogue takes aim...	1286	TED2006	1140739200	26	David Pogue	David Pogue: Simplicity sells	1	1151367060	[{'id': 7, 'name': 'Funny', 'count': 964}, {'i...	[{'id': 1725 'https://pe.tedcdn.
3	200	In an emotionally charged talk, MacArthur-winn...	1116	TED2006	1140912000	35	Majora Carter	Majora Carter: Greening the ghetto	1	1151367060	[{'id': 3, 'name': 'Courageous', 'count': 760}...	[{'id': 1041 'https://pe.tedcdn.
4	593	You've never seen data presented like this. Wi...	1190	TED2006	1140566400	48	Hans Rosling	Hans Rosling: The best stats you've ever seen	1	1151440680	[{'id': 9, 'name': 'Ingenious', 'count': 3202}...	[{'id': 2056 'https://pe.tedcdn.

```
# Размер датасета
data.shape
total_row = data.shape[0]
total_column = data.shape[1]
print('Всего строк: {}\nВсего столбцов: {}'.format(total_row, total_column))
```

```
Всего строк: 2550
Всего столбцов: 17
```

In [17]:
```
# Список колонок
columns = list(data.columns)
print(f'''Стобцы датасета: {', '.join(map(str, columns))}''')
```

```
Стобцы датасета: comments, description, duration, event, film_date, languages, main_speaker, name, num_speaker, published_date,
ratings, related_talks, speaker_occupation, tags, title, url, views
```

In [18]:
```
# Список колонок с типами данных
data.dtypes
```

Out[18]:
```
comments              int64
description          object
duration              int64
event                object
film_date             int64
languages             int64
main_speaker         object
name                 object
num_speaker           int64
published_date        int64
ratings              object
related_talks        object
speaker_occupation   object
tags                 object
title                object
url                  object
views                 int64
dtype: object
```

In [19]:
```
# Проверим наличие пустых значений
# Цикл по колонкам датасета
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))
```

```
comments - 0
description - 0
duration - 0
event - 0
film_date - 0
languages - 0
main_speaker - 0
name - 0
num_speaker - 0
published_date - 0
ratings - 0
related_talks - 0
speaker_occupation - 6
tags - 0
title - 0
url - 0
views - 0
```

In [20]:
```
# Основные статистические характеристки набора данных
data.describe()
```

Out[20]:

	comments	duration	film_date	languages	num_speaker	published_date	views
count	2550.000000	2550.000000	2.550000e+03	2550.000000	2550.000000	2.550000e+03	2.550000e+03
mean	191.562353	826.510196	1.321928e+09	27.326275	1.028235	1.343525e+09	1.698297e+06
std	282.315223	374.009138	1.197391e+08	9.563452	0.207705	9.464009e+07	2.498479e+06
min	2.000000	135.000000	7.464960e+07	0.000000	1.000000	1.151367e+09	5.044300e+04
25%	63.000000	577.000000	1.257466e+09	23.000000	1.000000	1.268463e+09	7.557928e+05
50%	118.000000	848.000000	1.333238e+09	28.000000	1.000000	1.340935e+09	1.124524e+06
75%	221.750000	1046.750000	1.412964e+09	33.000000	1.000000	1.423432e+09	1.700760e+06
max	6404.000000	5256.000000	1.503792e+09	72.000000	5.000000	1.506092e+09	4.722711e+07

In [21]:
```
# Определим уникальные значения для количества выступавших
data['num_speaker'].unique()
```

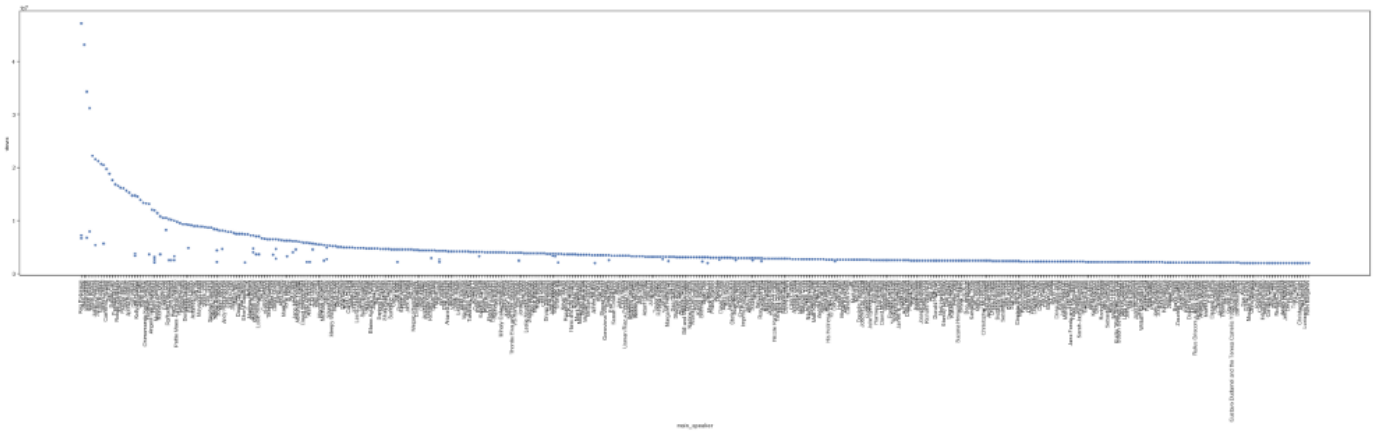
Out[21]:
```
array([1, 2, 3, 4, 5], dtype=int64)
```

3) *Визуальное исследование датасета*

- Диаграмма рассеяния

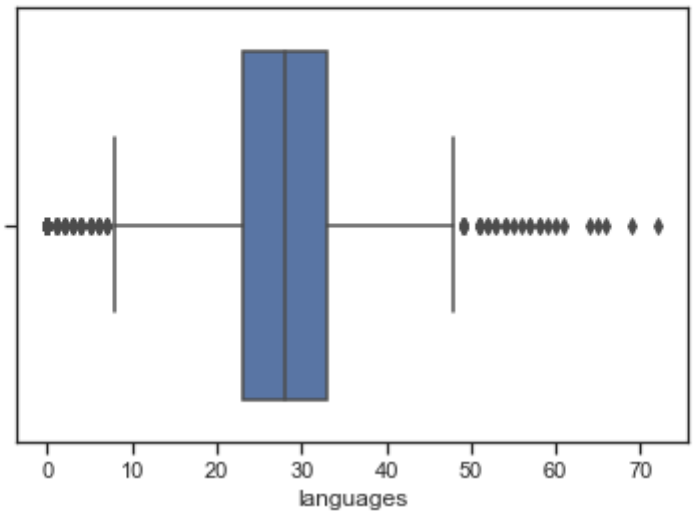
На данной диаграмме можно увидеть зависимость числа просмотров от спикера. График построен на основе отсортированного датасета по убыванию по просмотрам. Взяты первые 500 записей.

```
In [31]: #Диаграмма рассеяния
         fig, ax = plt.subplots(figsize=(50,10))
         plt.xticks(rotation=90)
         sns.scatterplot(ax=ax, x='main_speaker', y='views', data=data.sort_values(by='views', ascending=False)[:500])

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x1eb11288>
```



- Ящик с усами

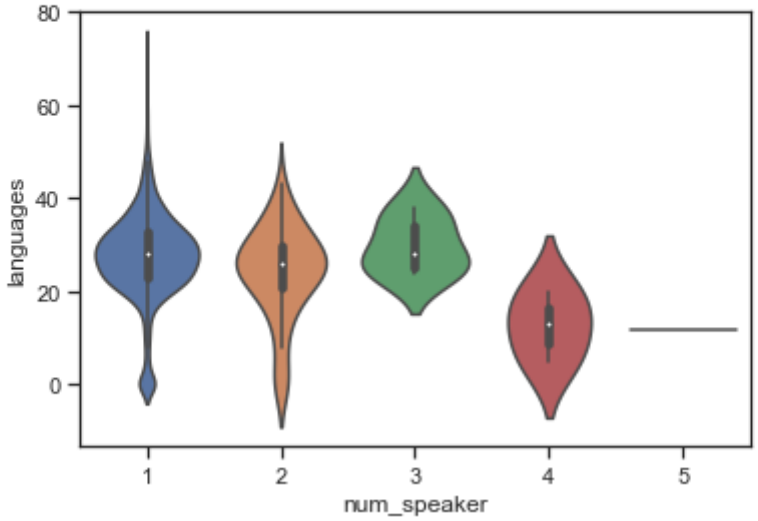
```
In [34]: #Ящик с усами
         sns.boxplot(x=data['languages'])

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x1fba34c8>
```



- Violin plot

In [36]:
```
# Распределение параметра languages сгруппированные по num_speaker.
sns.violinplot(x='num_speaker', y='languages', data=data)
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x1fc7ce88>



4) *Информация о корреляции признаков*

In [39]:
```
# Корреляционная матрица
data.corr()
```

Out[39]:

	comments	duration	film_date	languages	num_speaker	published_date	views
comments	1.000000	0.140694	-0.133303	0.318284	-0.035489	-0.185936	0.530939
duration	0.140694	1.000000	-0.242941	-0.295681	0.022257	-0.166324	0.048740
film_date	-0.133303	-0.242941	1.000000	-0.061957	0.040227	0.902565	0.006447
languages	0.318284	-0.295681	-0.061957	1.000000	-0.063100	-0.171836	0.377623
num_speaker	-0.035489	0.022257	0.040227	-0.063100	1.000000	0.049240	-0.026389
published_date	-0.185936	-0.166324	0.902565	-0.171836	0.049240	1.000000	-0.017920
views	0.530939	0.048740	0.006447	0.377623	-0.026389	-0.017920	1.000000

In [40]:
```
# визуализация корреляционной матрицы "тепловой" диаграммой
sns.heatmap(data.corr())
```

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1fde05c8>