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: **Ся Тунтун**

стулент группы:ИУ5И- 23М

Москва 2022г.

Цель работы:

1, изучение различных методов визуализация данных и создание истории на основе данных.

Задание:

- Выбрать набор данных (датасет).

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

1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
4. Выбор графиков должен быть обоснован использованием методологии data-to-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.
5. История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.

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

1 Обзор

Рынок цен Google

Набор данных содержит 500 лучших приложений для android, доступных в магазине google play для следующих категорий: Все категории, Искусство и дизайн, Авто и транспортные средства, Красота, Книги и справочники, Бизнес, Комиксы, Общение, Образование, Развлечения, События, Финансы, Еда и напитки, Здоровье и фитнес, Дом и быт, Библиотеки и демо, Стиль жизни, Карты и навигация, Медицина, Музыка и аудио, Новости и журналы, Воспитание, Персонализация, Фотография, Продуктивность, Покупки, Социальные, Спорт, Инструменты, Путешествия и местное, Видеоплееры и редакторы.

Рейтинг приложений основан на рейтинге приложений в магазине google play store за январь 2022 года.

2,Конкретный процесс и реализация кода

In [1]:
```
"""
/kaggle/input/google-play-store-category-wise-top-500-apps/Business.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Food  Drink.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Shopping.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Parenting.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Auto  Vehicles.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/All Categories.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/News  Magazines.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Comics.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Art  Design.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Medical.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Lifestyle.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Music  Audio.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Libraries  Demo.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Finance.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Video Players  Editors.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Maps  Navigation.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Beauty.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Travel  Local.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Sports.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Personalisation.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Social.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Tools.csv
/kaggle/input/google-play-store-category-wise-top-500-apps/Education.csv
"""
```

Out[1]:
```
'\n/kaggle/input/google-play-store-category-wise-top-500-apps/Business.csv\n/kaggle/input/go
ogle-play-store-category-wise-top-500-apps/Food  Drink.csv\n/kaggle/input/google-play-store-
category-wise-top-500-apps/Shopping.csv\n/kaggle/input/google-play-store-category-wise-top-5
00-apps/Parenting.csv\n/kaggle/input/google-play-store-category-wise-top-500-apps/Auto  Vehi
cles.csv\n/kaggle/input/google-play-store-category-wise-top-500-apps/All Categories.csv\n/ka
ggle/input/google-play-store-category-wise-top-500-apps/News  Magazines.csv\n/kaggle/input/g
oogle-play-store-category-wise-top-500-apps/Comics.csv\n/kaggle/input/google-play-store-cate
gory-wise-top-500-apps/Art  Design.csv\n/kaggle/input/google-play-store-category-wise-top-50
0-apps/Medical.csv\n/kaggle/input/google-play-store-category-wise-top-500-apps/Lifestyle.csv
\n/kaggle/input/google-play-store-category-wise-top-500-apps/Music  Audio.csv\n/kaggle/inpu
t/google-play-store-category-wise-top-500-apps/Libraries  Demo.csv\n/kaggle/input/google-pla
y-store-category-wise-top-500-apps/Finance.csv\n/kaggle/input/google-play-store-category-wis
e-top-500-apps/Video Players  Editors.csv\n/kaggle/input/google-play-store-category-wise-top
-500-apps/Maps  Navigation.csv\n/kaggle/input/google-play-store-category-wise-top-500-apps/B
eauty.csv\n/kaggle/input/google-play-store-category-wise-top-500-apps/Travel  Local.csv\n/ka
ggle/input/google-play-store-category-wise-top-500-apps/Sports.csv\n/kaggle/input/google-pla
```

Импорт данных

In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot
from plotly.offline import init_notebook_mode, iplot, plot
import plotly as py
init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.express as px
```

Я создаю функцию для чтения файла, потому что иногда, когда вы получаете ошибку, чтобы исправить ее, вам может понадобиться получить исходный набор данных, поэтому вызов функции намного проще.

In [3]:

```
def getDf():

    df = pd.read_csv('/kaggle/input/google-play-store-category-wise-top-500-apps/All Categories.csv')
    return df


df = getDf()
```

Общий вид набора данных

In [4]:

```
df.head()
```

Out[4]:

	Rank	Name	Developer	Category	Size	Star Rating	Reviews	Downloads	Rated for
0	1	Meesho: Online Shopping App	Meesho	Shopping	15 MB	4.4	15L	10Cr+	3+
1	2	Shopee: Online Shopping	Shopee	Shopping	68 MB	4.1	76T	1Cr+	3+
2	3	Instagram	Instagram	Social	41 MB	4.3	13Cr	100Cr+	12+
3	4	MX Player: Videos, OTT & Games	MX Media (formerly J2 Interactive)	Video Players & Editors	36 MB	4.1	1Cr	100Cr+	3+
4	5	speedfiy	PRIME DIGITAL PTE. LTD.	Tools	12 MB	4.5	41T	1Cr+	3+

In [5]:
```
df.describe()
```

Out[5]:

	Rank	Star Rating
count	600.000000	599.000000
mean	300.500000	4.156427
std	173.349358	0.362896
min	1.000000	2.100000
25%	150.750000	4.000000
50%	300.500000	4.200000
75%	450.250000	4.400000
max	600.000000	4.900000

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 9 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Rank         600 non-null    int64  
 1   Name         600 non-null    object 
 2   Developer    600 non-null    object 
 3   Category     600 non-null    object 
 4   Size         600 non-null    object 
 5   Star Rating  599 non-null    float64
 6   Reviews      599 non-null    object 
 7   Downloads    600 non-null    object 
 8   Rated for    600 non-null    object 
dtypes: float64(1), int64(1), object(7)
memory usage: 42.3+ KB
```

Мы можем изменить тип этих колонок для создания сюжета

Reviews Column

Downloads Column

Size Column

Анализ данных

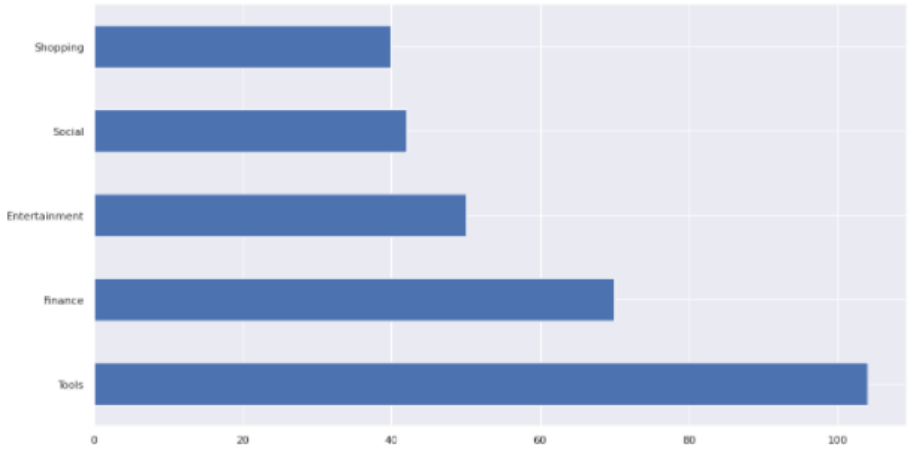
Get the most used 5 categories

In [7]:

```
sns.set(rc = {'figure.figsize':(15,8)})

df['Category'].value_counts()[:5].plot(kind='barh')
```

Out[7]:

<AxesSubplot:>



Get the top 15 best Application

In [8]:

```
top15 = df[['Name', 'Rank','Category']].head(15)
top15
```

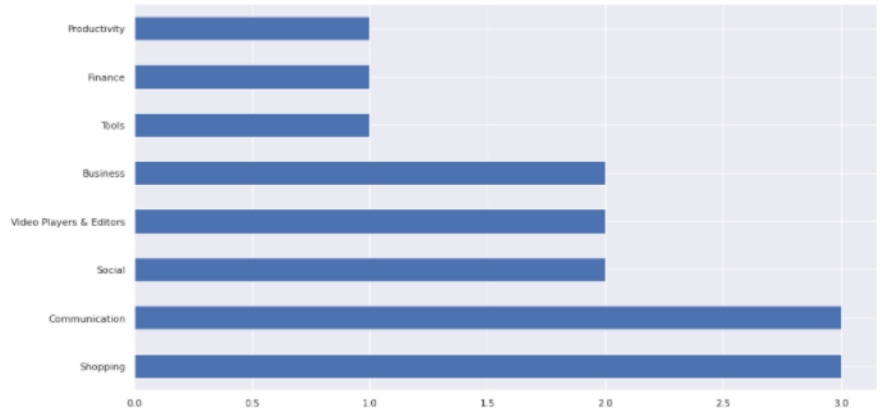
Out[8]:

	Name	Rank	Category
0	Meesho: Online Shopping App	1	Shopping
1	Shopee: Online Shopping	2	Shopping
2	Instagram	3	Social
3	MX Player: Videos, OTT & Games	4	Video Players & Editors
4	speedfiy	5	Tools
5	Snapchat	6	Communication
6	ZOOM Cloud Meetings	7	Business
7	Flipkart Online Shopping App	8	Shopping
8	Telegram	9	Communication
9	Chingari - powered by GARI	10	Social
10	mAst: Music Status Video Maker	11	Video Players & Editors
11	Google Meet	12	Business
12	PhonePe: UPI, Recharge, Investment, Insurance	13	Finance
13	Truecaller: Caller ID & Block	14	Communication
14	MyJio: For Everything Jio	15	Productivity

Мы видим, что в топ-15 самых рейтинговых приложений есть только одно приложение категории Инструменты

In [9]:
```
sns.set(rc = {'figure.figsize':(15,8)})

top15['Category'].value_counts().plot(kind='barh')
```

Out[9]:
```
<AxesSubplot:>
```



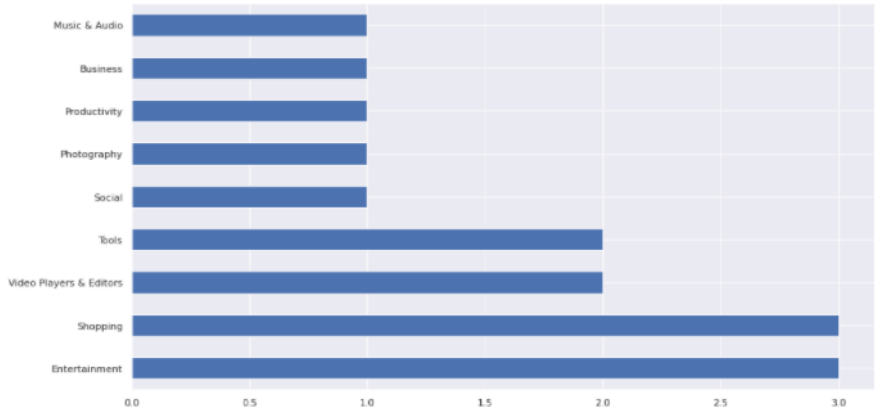
Приложение для шопинга занимает достойное место в рейтинге

Первая версия

Мы видим, что в топ-15 самых рейтинговых приложений есть только одно приложение категории Инструменты

In [10]:
```
sns.set(rc = {'figure.figsize':(15,8)})

df['Category'].iloc[-15:].value_counts().plot(kind='barh')
```

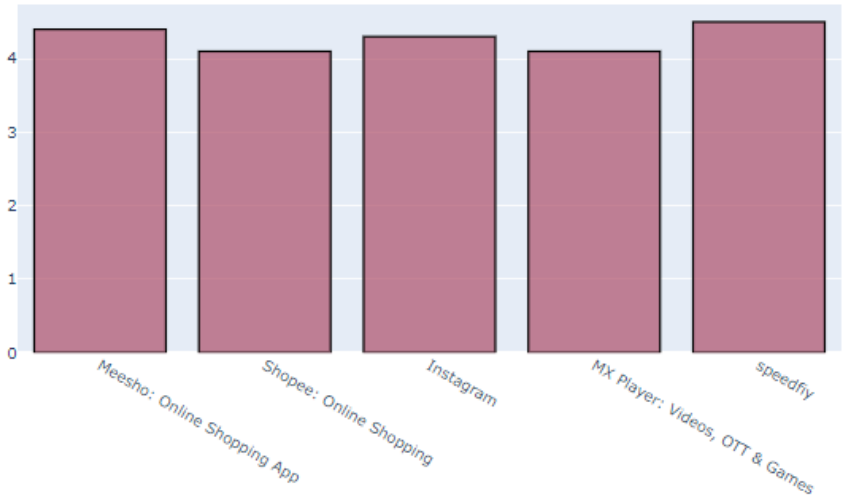
Out[10]:
```
<AxesSubplot:>
```



Рейтинг пяти лучших приложений

In [11]:
```
#sns.barplot(x=df['Name'].head() ,y=df['Star Rating'].head(),palette='flare')

trace1 = go.Bar(
                x = df['Name'].head(),
                y = df['Star Rating'].head(),
                name = "Star Rating",
                marker = dict(color = 'rgba(174, 80, 107, 0.7)',
                              line=dict(color='rgb(0,0,0)',width=1.5)))

data = [trace1]
layout = {
  'xaxis': {'title': 'Top 5 Ranked Application'},
  'title': 'The Score of the top 5 best Application',
  'barmode': 'relative'
}
fig = go.Figure(data = data, layout = layout)
iplot(fig)
```

оценка 5 лучших приложений



Измените тип данных Downloads и Review, чтобы сделать Analyse

In [12]:
```
df = getDf()

def changeDownloadsType():

    for i in range(len(df['Downloads'])):

        if 'Cr+' in df['Downloads'][i]:
            if 'T' in df['Downloads'][i]:
                df['Downloads'][i] = 10000000000

            else:
                intValue = int(df['Downloads'][i][:-3])
                df['Downloads'][i] = intValue * 10000000

        elif 'L' in df['Downloads'][i] or 'T' in df['Downloads'][i]:
            intValue = int(df['Downloads'][i][:-2])
            df['Downloads'][i] = intValue * 100000
```

```
    df['Downloads'] =  df['Downloads'].astype(int)
    return df



df = changeDownloadsType()
```

/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:13: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:17: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:9: SettingWithCopyWarning:

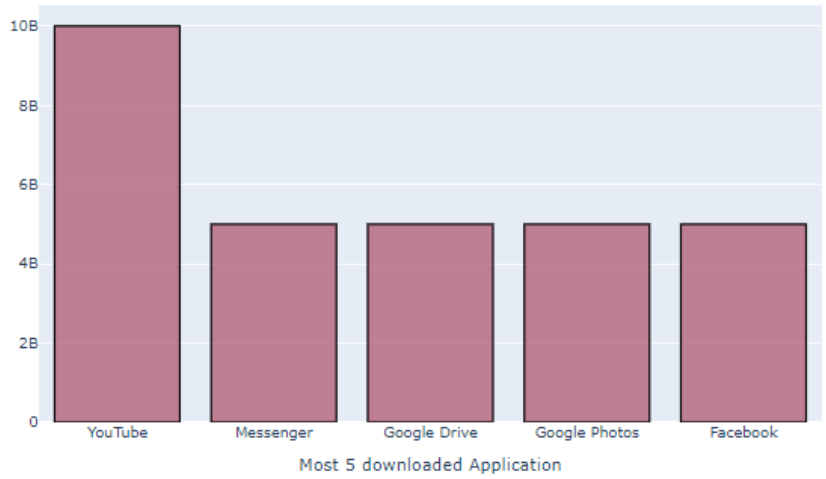
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

In [13]:

```
#sns.barplot(x=df['Name'].head() ,y=df['Star Rating'].head(),palette='flare')

mostDownloadedApp = df[['Name','Downloads']].sort_values(by='Downloads', ascending=False)

trace1 = go.Bar(
                x = mostDownloadedApp['Name'].head(),
                y = mostDownloadedApp['Downloads'].head(),
                name = "Downloads",
                marker = dict(color = 'rgba(174, 80, 107, 0.7)',
                              line=dict(color='rgb(0,0,0)',width=1.5)))

data = [trace1]
layout = {
  'xaxis': {'title': 'Most 5 downloaded Application'},
  'barmode': 'relative'
}
fig = go.Figure(data = data, layout = layout)
iplot(fig)
```

```
fig = px.scatter(x=df['Downloads'] , y=df['Star Rating'],title='Downloads vs Rating')

fig.show()
```



загрузка против рейтинга