ИУ5-62Б Дума Эмилия Михайловна РК1 ТМО

Технологии разведочного анализа и обработки данных.

Вариант 8

Задача 1, датасет 8.

Для заданного набора данных проведите корреляционный анализ. В случае наличия пропусков в данных удалите строки или колонки, содержащие пропуски. Сделайте выводы о возможности построения моделей машинного обучения и о возможном вкладе признаков в модель.

Датасет <https://www.kaggle.com/lava18/google-play-store-apps>

Задание для студентов групп ИУ5-62Б, ИУ5Ц-82Б - для произвольной колонки данных построить гистограмму.

#!/usr/bin/env python

# coding: utf-8

# In[2]:

# ИУ5-62Б Дума РК1

import numpy as np

import pandas as pd

import seaborn as sns

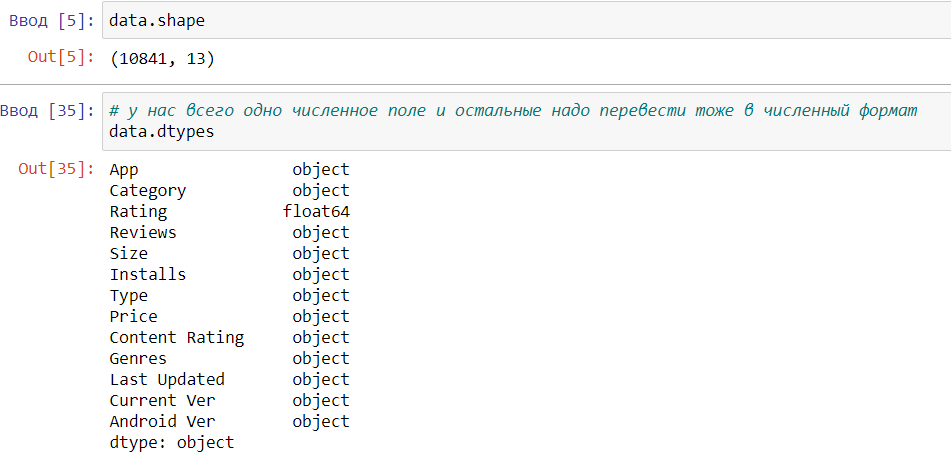
import matplotlib.pyplot as plt

get\_ipython().run\_line\_magic('matplotlib', 'inline')

sns.set(style="ticks")

# In[4]:

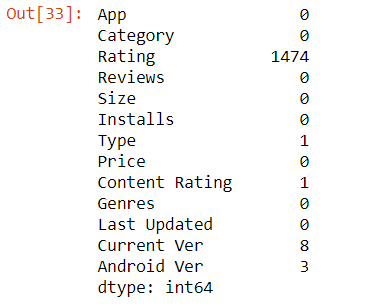
data = pd.read\_csv('googleplaystore.csv', sep=",")



# In[33]:

#видно пропуски -> датасет непригоден

data.isnull().sum()



# In[8]:

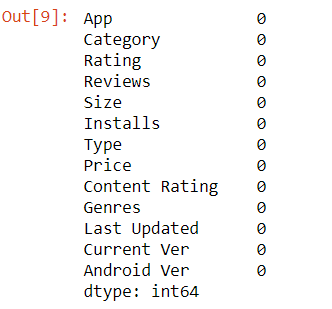
# удаляем пропуски

data\_new = data.dropna(axis=0, how='any')

print(f"{data.shape[0]-data\_new.shape[0]} rows were deleted")

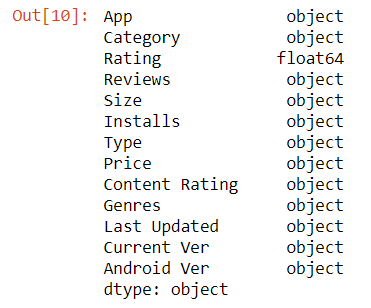
# In[9]:

data\_new.isnull().sum()



# In[10]:

data\_new.dtypes



# In[11]:

data\_new.head(10)



# In[12]:

pd.options.mode.chained\_assignment = None

# In[13]:

data\_new.loc[:, 'Reviews']=pd.to\_numeric(data\_new.loc[:, 'Reviews'])

# In[14]:

data\_new.loc[:, 'Installs'] = data\_new.loc[:, 'Installs'].str.replace(',','')

data\_new.loc[:, 'Installs'] = data\_new.loc[:, 'Installs'].map(lambda x: str(x).rstrip('+'))

data\_new.loc[:, 'Installs'] = pd.to\_numeric(data\_new.loc[:, 'Installs'], errors='coerce', downcast='float')

# In[15]:

data\_new.loc[:, 'Price'] = data\_new.loc[:, 'Price'].map(lambda x: str(x).lstrip('$'))

data\_new.loc[:, 'Price'] = pd.to\_numeric(data\_new.loc[:, 'Price'], errors='coerce', downcast='float')

# In[16]:

data\_new.loc[:, 'Size'] = data\_new.loc[:, 'Size'].str.replace(',','')

data\_new.loc[:, 'Size'] = data\_new.loc[:, 'Size'].map(lambda x: str(x).rstrip('M'))

# In[18]:

from sklearn.impute import SimpleImputer

from sklearn.impute import MissingIndicator

# In[19]:

data\_new.loc[:, 'Size'] = pd.to\_numeric(data\_new.loc[:, 'Size'], errors='coerce', downcast='float')

temp\_data = data\_new[['Size']]

indicator = MissingIndicator()

mask\_missing\_values\_only = indicator.fit\_transform(temp\_data)

imp\_num = SimpleImputer(strategy='mean')

data\_num\_imp = imp\_num.fit\_transform(temp\_data)

filled\_data = data\_num\_imp[mask\_missing\_values\_only]

data\_new = data\_new.replace(np.nan, filled\_data[0])

# In[20]:

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

le = LabelEncoder()

cat\_enc\_le = le.fit\_transform(data\_new.loc[:, 'Category'])

print(le.inverse\_transform(np.unique(cat\_enc\_le)))

data\_new.loc[:, 'Category'] = cat\_enc\_le

['ART\_AND\_DESIGN' 'AUTO\_AND\_VEHICLES' 'BEAUTY' 'BOOKS\_AND\_REFERENCE'

'BUSINESS' 'COMICS' 'COMMUNICATION' 'DATING' 'EDUCATION' 'ENTERTAINMENT'

'EVENTS' 'FAMILY' 'FINANCE' 'FOOD\_AND\_DRINK' 'GAME' 'HEALTH\_AND\_FITNESS'

'HOUSE\_AND\_HOME' 'LIBRARIES\_AND\_DEMO' 'LIFESTYLE' 'MAPS\_AND\_NAVIGATION'

'MEDICAL' 'NEWS\_AND\_MAGAZINES' 'PARENTING' 'PERSONALIZATION'

'PHOTOGRAPHY' 'PRODUCTIVITY' 'SHOPPING' 'SOCIAL' 'SPORTS' 'TOOLS'

'TRAVEL\_AND\_LOCAL' 'VIDEO\_PLAYERS' 'WEATHER']

# In[21]:

cat\_enc\_le = le.fit\_transform(data\_new.loc[:, 'Content Rating'])

print(le.inverse\_transform(np.unique(cat\_enc\_le)))

data\_new.loc[:, 'Content Rating'] = cat\_enc\_le

['Adults only 18+' 'Everyone' 'Everyone 10+' 'Mature 17+' 'Teen' 'Unrated']

# In[22]:

cat\_enc\_le = le.fit\_transform(data\_new.loc[:, 'Type'])

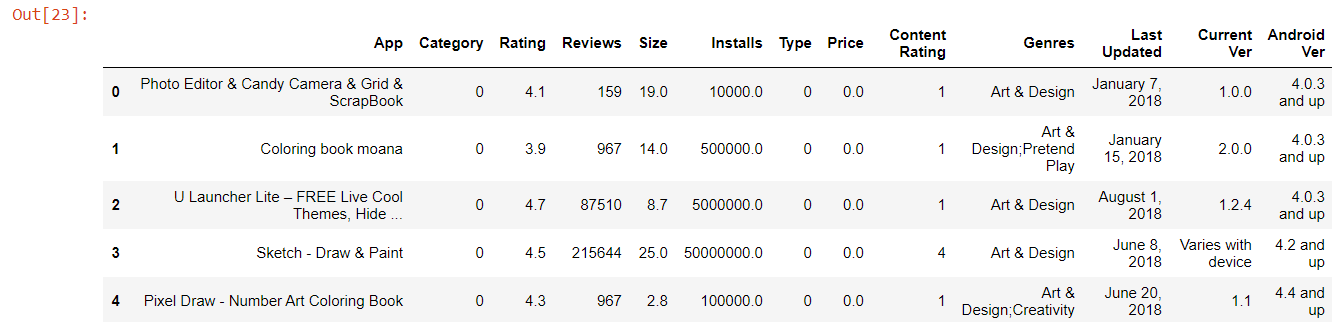
print(le.inverse\_transform(np.unique(cat\_enc\_le)))

data\_new.loc[:, 'Type'] = cat\_enc\_le

['Free' 'Paid']

# In[23]:

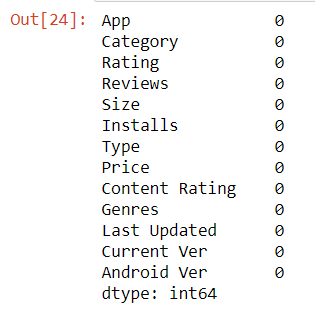
data\_new.head()



# In[24]:

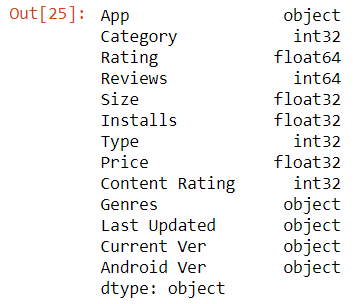
# теперь все строки без пропусков и в численном формате

data\_new.isnull().sum()



# In[25]:

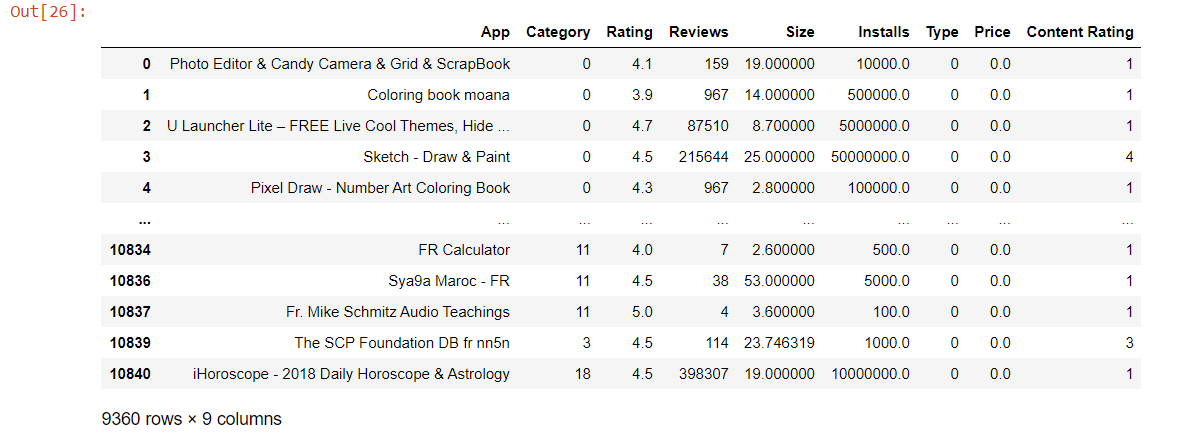
data\_new.dtypes



# In[26]:

# уберём колонки (см. в квадратных скобках) т.к. они не имеют особого значения, жанры - колонка с множеством значений

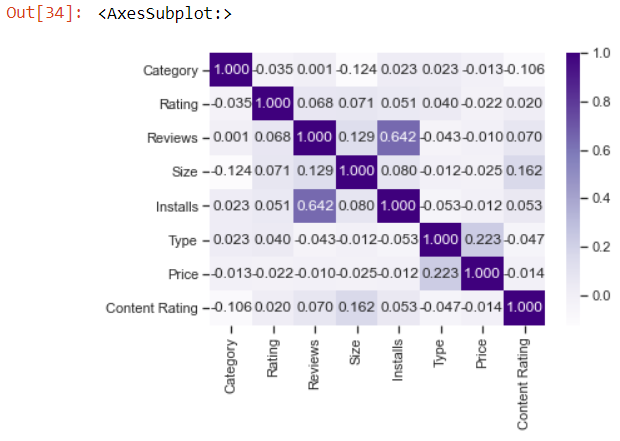
data\_new.drop(columns=['Genres', 'Last Updated', 'Current Ver', 'Android Ver'])



# In[34]:

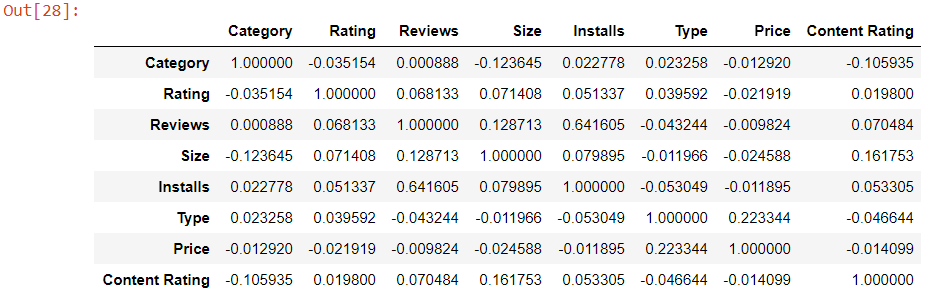
# проведём корреляционный анализ

sns.heatmap(data\_new.corr(), cmap='Purples', annot=True, fmt='.3f')



# In[28]:

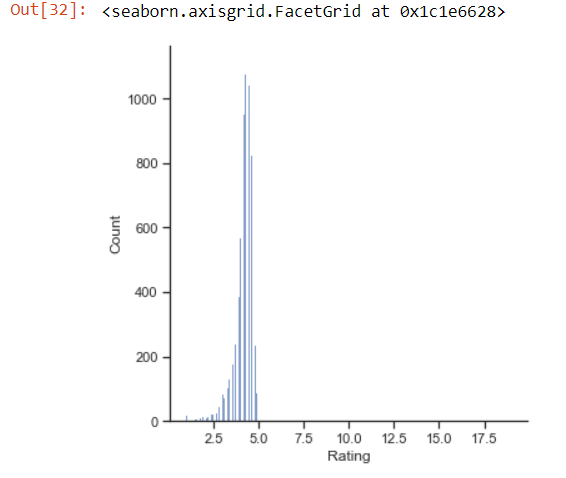
data\_new.corr()



# In[32]:

# построим гистограмму по столбцу Rating

sns.displot(x=data['Rating'])



**По результатам кореляционного анализа и гистограмме можно сделать выводы:**

1 Средняя оценка приложения ~ 4.6

2 Параметр Size слабо связан с параметрами Category и Reviews - это видно и на диаграммах, и по значению в районе 0.1

3 Параметр Type слабо корелирует с параметром Price (очевидно)

Вывод: в модели данных существует всего одна сильная кореляция, для Installations-Reviews. Построить хорошую модель будет сложно, т.к. большинство параметров не корелирует между собой