НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

з дисципліни «Математичне моделювання»

на тему

**Моделювання системи для прогнозування популярності мобільних пристроїв**

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-83 | *Норкін Б.В.* |
| *Макаренко А.С.* |  |

Київ — 2021

## Вступ

## Постановка задачі

Предметною областю даного дослідження буде сфера продажу мобільних пристороїв. Вона є актуальною, адже майже в кожної людини є мобільний телефон. Основна задача - це **прогнозування популярності мобільного пристрою за його параметрами** (об'єм пам'яті, розмір екрану, операційною системою тощо). Результати дослідження будуть корисними для корпорацій-виробників мобільних телефонів, які бажають дізнатися чи буде користуватися попитом та чи інша модель телефону. На даному етапі основна ціль - підготувати датасет до можливості побудови математичної моделі на його основі. Це передбачає завантаження даних, їхню очистку, редагування.

В даній роботі було обрано задачу лассо-регресії. Було взято датасет "Mobile Phones Data". Задачею моделювання в рамках конкретного контексту є моделювання системи для передбачення популярності мобільного пристрою.

**Огляд літератури**

1. <https://pythobyte.com/lasso-regression-73016c58/>
2. The Elements of Statistical Learning Data Mining/ Inference/and

Prediction/Trevor Hastie Robert Tibshirani Jerome Friedman

1. /python-data-science-machine-learning-tutorial/
2. <https://uaspectr.com/2021/01/05/top-brendiv-smartfoniv-v-ukrayini-ta-sviti-2020/>
3. http://www.machinelearning.ru/wiki/

# Проектування математичного забезпечення

**Перелік методів розв'язання задачі**

З точки зору статистики та машинного навчання, задача прогнозування є задачею регресії. Для розв'язання задачі регресії необхідно змоделювати взаємоз'язок між змінними (залежними та незалежними). Система повинна прогнозувати популярність мобільного телефону, а отже, необхідно знайти зв'язок між його параметрами.

Для розв'язання задачі регресії розроблені алгоритми спеціальні алгоритми:

-**Лінійна Регресія**

**-Лассо-регресія**

**-Логістична Регресія**

**-Поліноміальна Регресія**

**-Дерева Прийняття Рішень**

**Лінійна регресія** будує логістичну криву, яка вираховує ймовірність виникнення певної події, саме тому вона більше підходить для задач класифікації. З плюсів можна виділити швидке моделювання, а серед мінусів - складність будування поліноміальних моделей.

**Лассо-регресія** - ще аналог лінійної регресії. Дозволяє позбутися від зайвих незалежних змінних, які не впливають на залежну змінну.

**Логістична регресія** використовується, коли залежна змінна є двійковою, тому цей метод більше підходить до задач класифікації, ніж до задачі регресії.

**Поліноміальна регресія** будує криву довільного порядку (квадратичну, кубічну і тд).

**Дерево рішень** - простий для розуміння алгоритм. Щоб змоделювати значення залежної змінної, перевіряються різні умови, які за своєю структурою нагадують дерево. Кінцевим результатом є дерево із вузлами прийняття рішень (decision nodes) та листя (leafs). На ребрах («гілках») дерева рішення записані ознаки, від яких залежить цільова функція, а в «листі» записані значення цільової функції, а в інших вузлах - ознаки, за якими розрізняються випадки. Щоб класифікувати новий випадок, треба спуститися по дереву до листа і видати відповідне значення.

Для вирішення даної задачі, було обрано метод лассо-регресії. Цей алгоритм легко інтерпретувати (тобто зрозуміти, що саме намагається зробити цей алгоритм). Було виявлено змінні, які майже не впливають на рейтинг телефонів, наприклад найнижча ціна.

**Контрольний приклад**

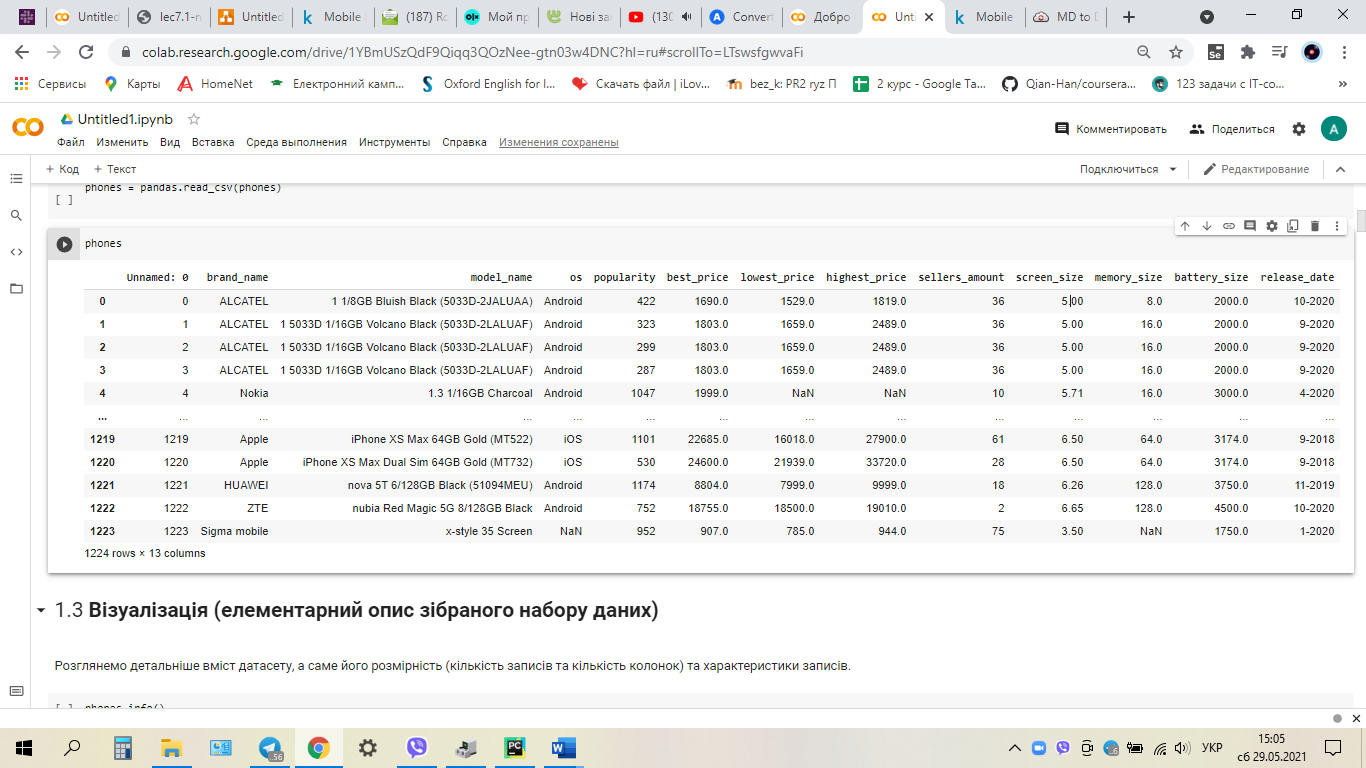
Для виконання цього завдання буде використано датасет, який був завантажений на веб-сайті [Kaggle](https://www.kaggle.com/artempozdniakov/ukrainian-market-mobile-phones-data). Він містить дані про мобільні телефони, випущені за останні 4 роки та які можна купити в Україні.

Для подальшої роботи з датасетом, скористаємося бібліотекою pandas. Зчитати csv-файл можна за допомогою pandas.read\_csv.

*import pandas   
from sklearn.model\_selection import train\_test\_split  
import seaborn as sns  
import matplotlib.pyplot as plt  
import matplotlib.mlab as mlab  
import matplotlib  
import numpy as np  
plt.style.use('ggplot')  
from matplotlib.pyplot import figure  
phones = 'drive/MyDrive/phones\_data.csv'*

*phones = pandas.read\_csv(phones)*

*phones*

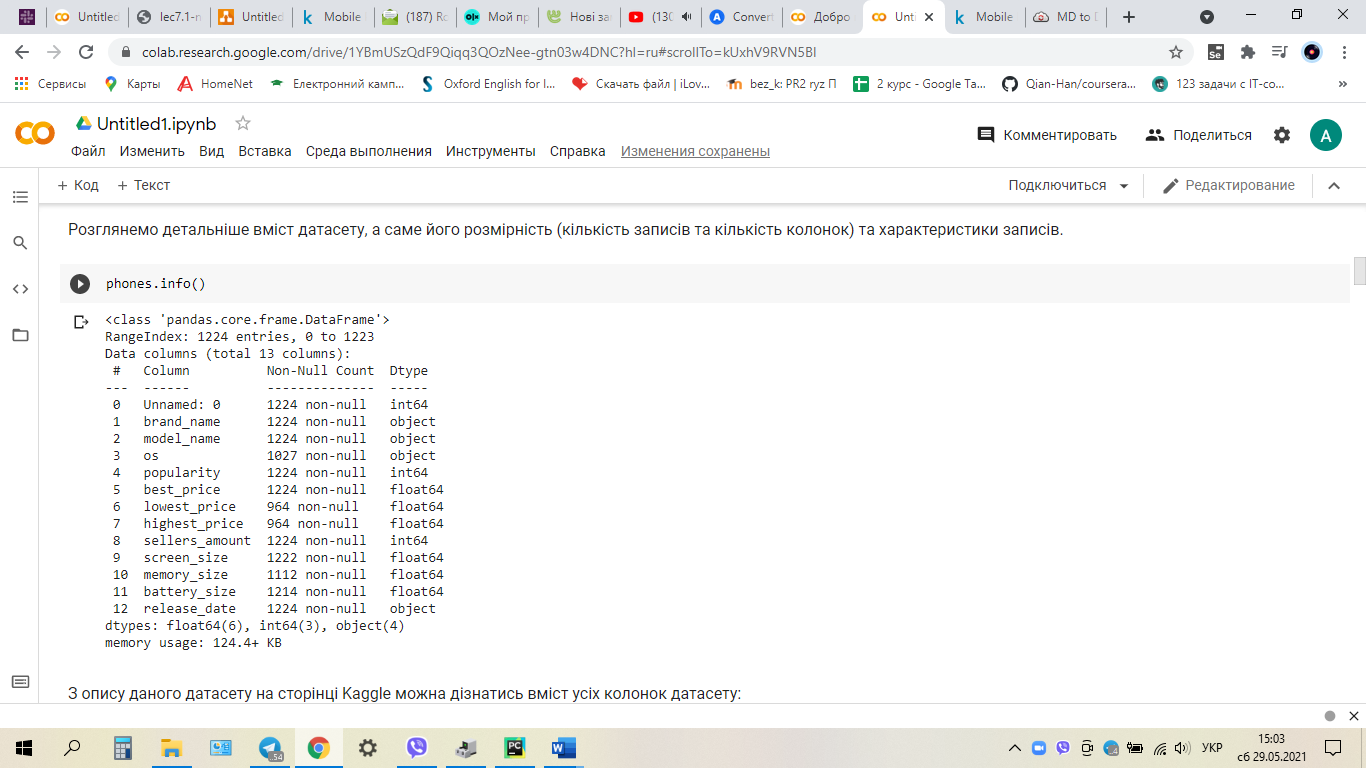


# 

# Візуалізація (елементарний опис зібраного набору даних)

Розглянемо детальніше вміст датасету, а саме його розмірність (кількість записів та кількість колонок) та характеристики записів.

*phones.info()*



З опису даного датасету на сторінці Kaggle можна дізнатись вміст усіх колонок датасету:

***brand\_name***- назва бренду телефону;

***model\_name*** - назва моделі мобільного пристрою;

***os*** - операційна система;

***popularity*** - популярність;

***best\_price*** - найкраща ціна;

***lowest\_price*** - найнижча ціна;

***highest\_price*** - найвища ціна;

***sellers\_amount*** - кількість продавців;

***screen\_size*** - розмір екрану;

***memory\_size*** - розмір пам'яті пристрою;

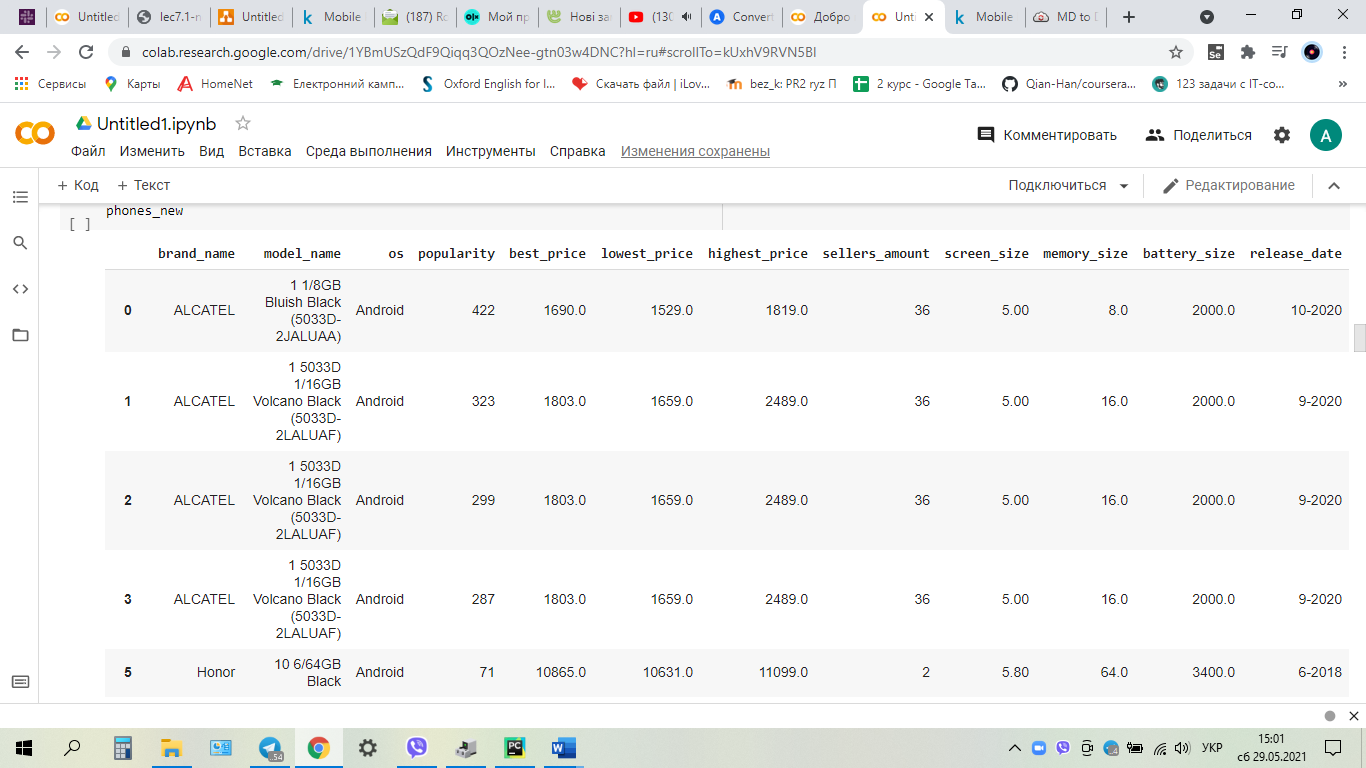
***battery\_size*** - розмір батареї;

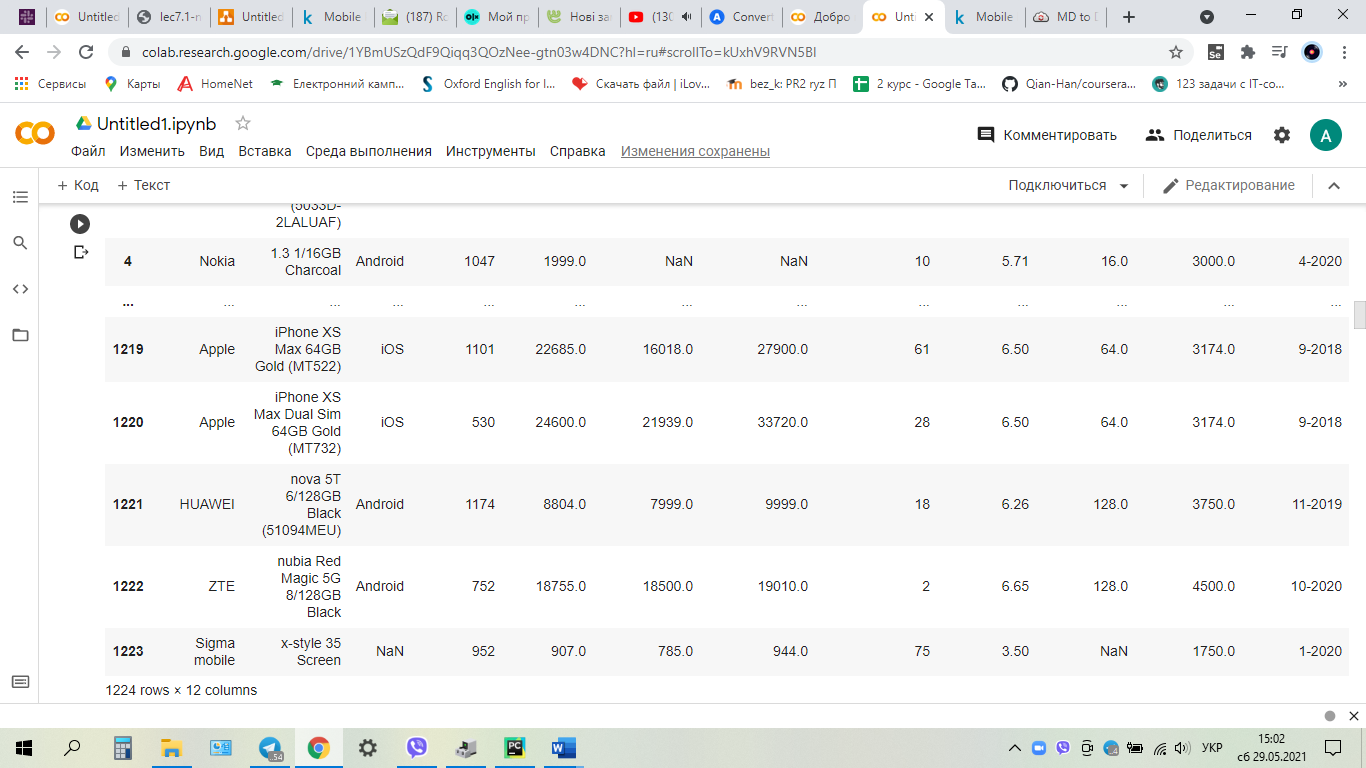
***release\_date*** - дата випуску.

# Первинна очистка даних

Оскільки для нашої таблиці вже існує стовпчик з порядковим номером, можна видалити стовпець unnamed:

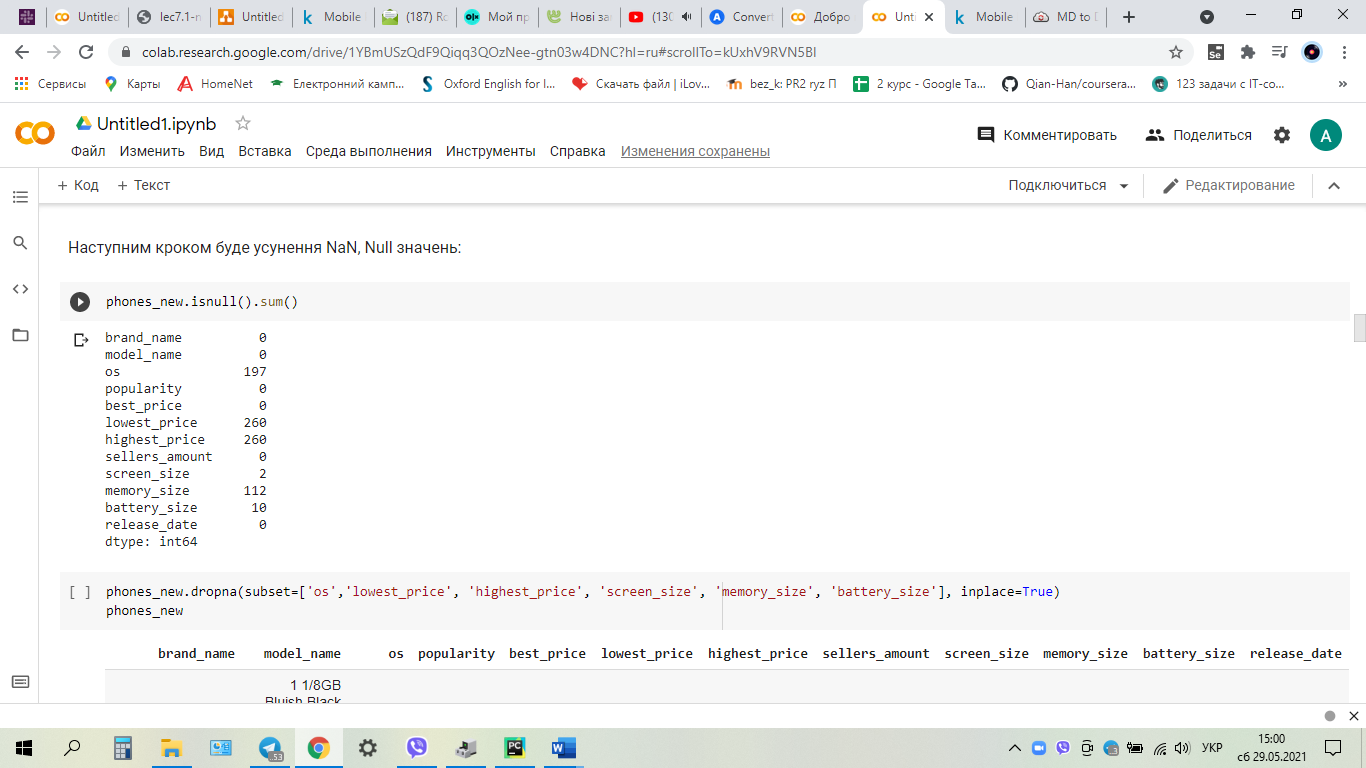
*phones\_new = phones.drop(columns="Unnamed: 0")  
phones\_new*





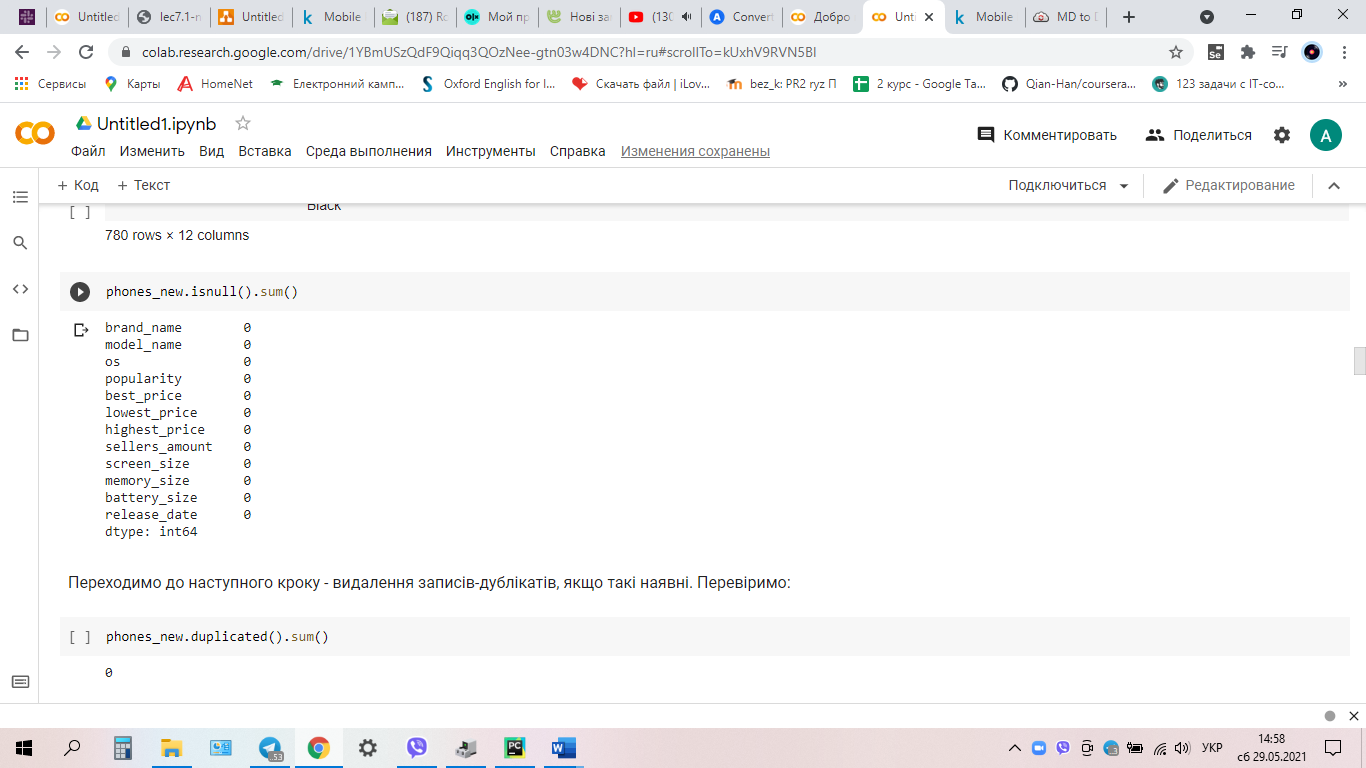
Наступним кроком буде усунення NaN, Null значень:

*phones\_new.isnull().sum()*

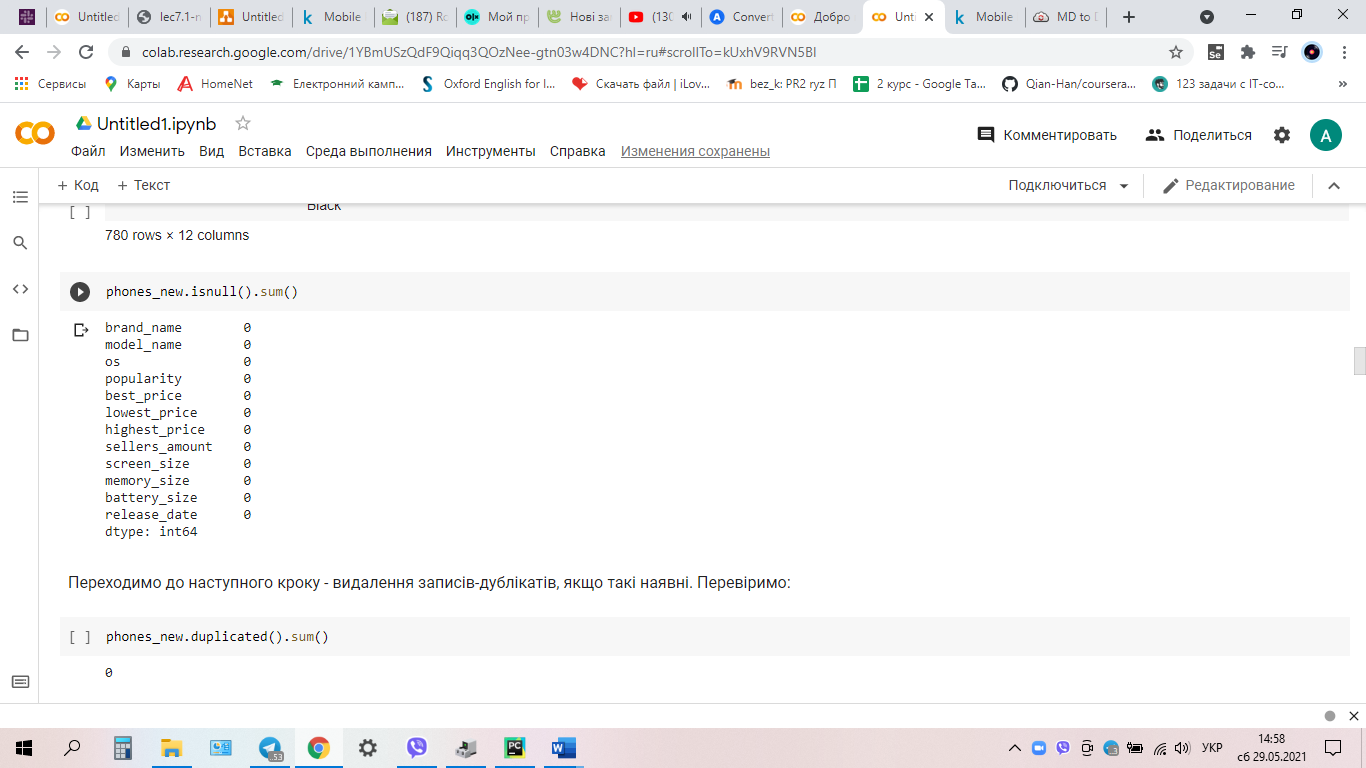


*phones\_new.dropna(subset=['os','lowest\_price', 'highest\_price', 'screen\_size', 'memory\_size', 'battery\_size'], inplace=True)  
phones\_new*

*phones\_new.isnull().sum()*



Як бачимо, NaN, Null значення в датасеті тепер відсутні. Переходимо до наступного кроку - видалення записів-дублікатів, якщо такі наявні. Перевіримо:

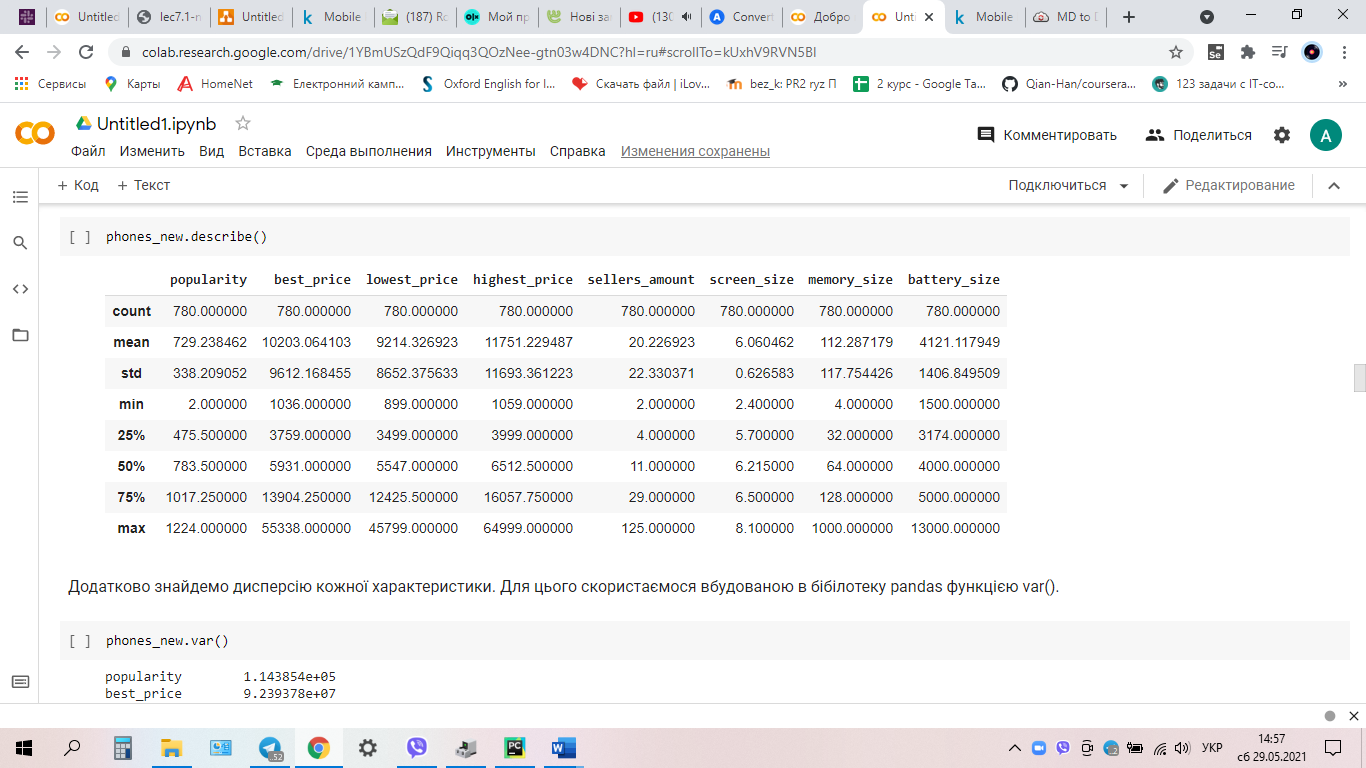


Отже, записи-дублікати в датасеті відсутні.

# Розвідувальний аналіз даних (EDA)

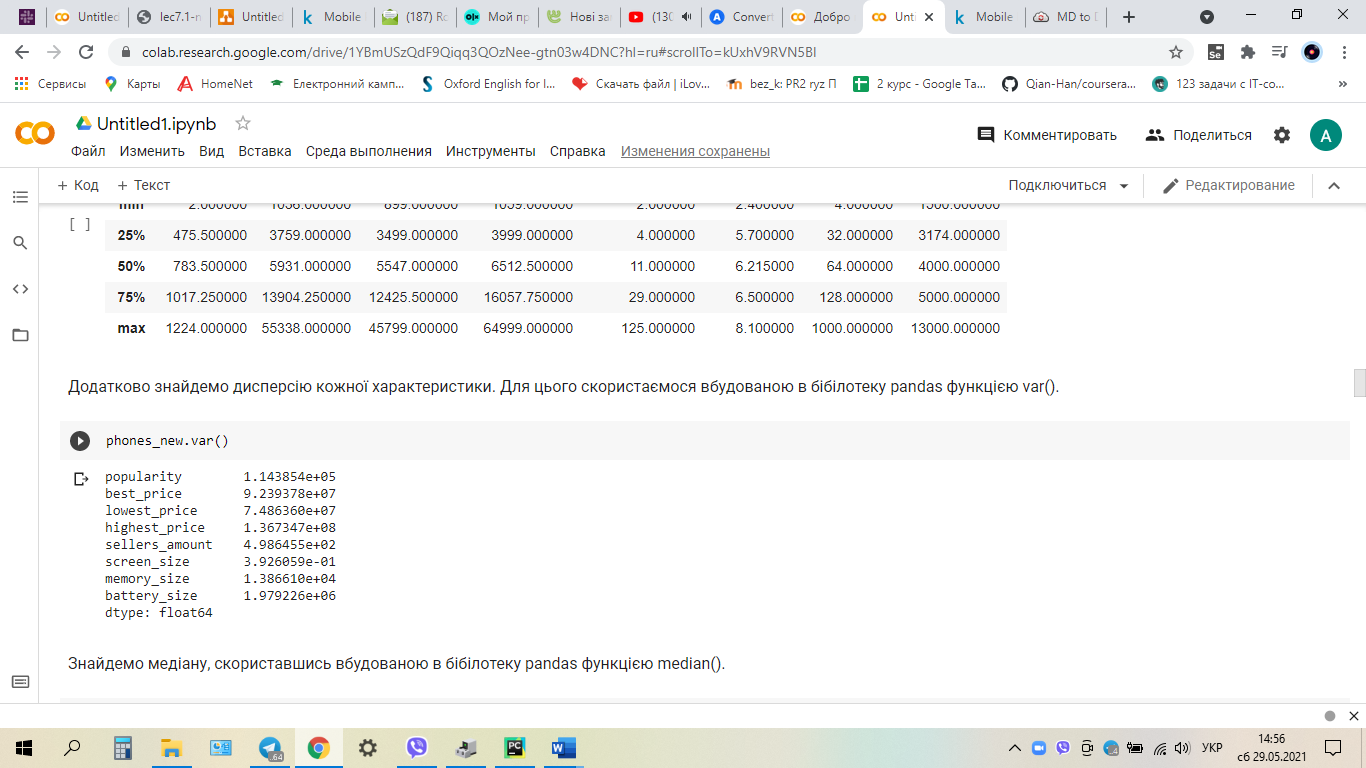
Застосуємо описову статистику: знайдемо кількість non-NA/null значень, середні арифметичні, стандартні відхилення, мінімальне та максимальне значення кожної характеристики, скориставшись функцією describe():

*phones\_new.describe()*



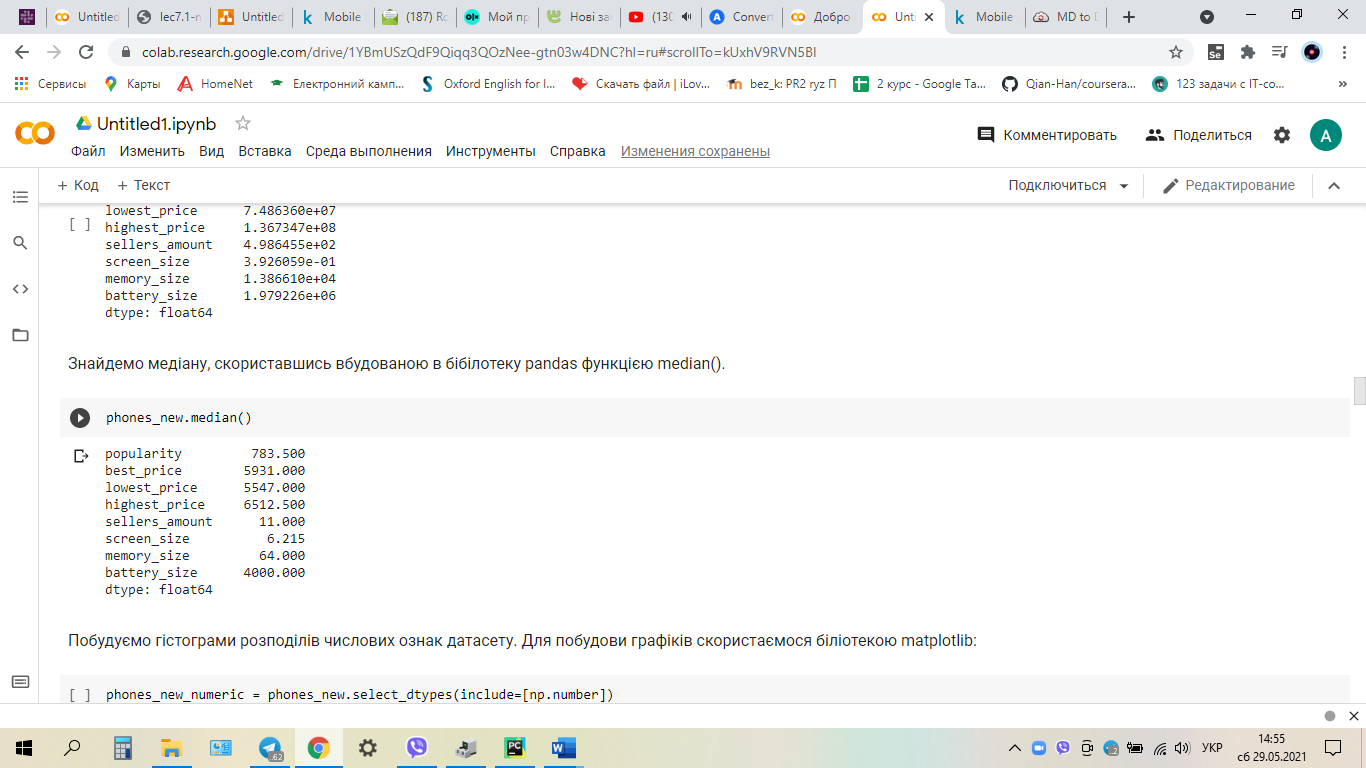
Додатково знайдемо дисперсію кожної характеристики. Для цього скористаємося вбудованою в бібілотеку pandas функцією var().

*phones\_new.var()*



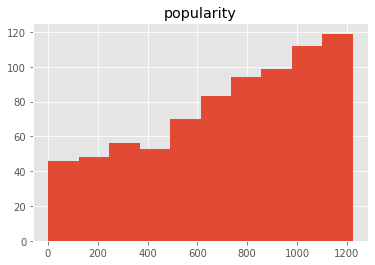
Знайдемо медіану, скориставшись вбудованою в бібілотеку pandas функцією median().

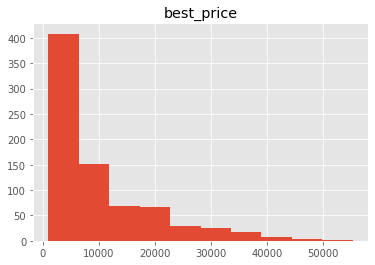
*phones\_new.median()*

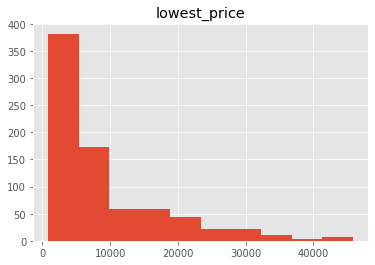


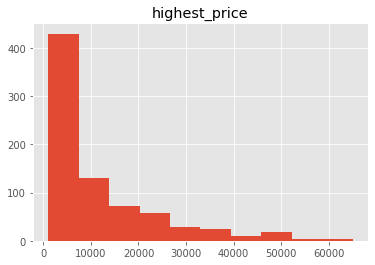
Побудуємо гістограми розподілів числових ознак датасету. Для побудови графіків скористаємося біліотекою matplotlib:

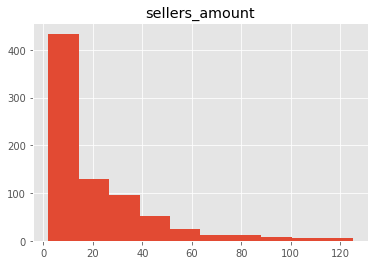
*phones\_new\_numeric = phones\_new.select\_dtypes(include=[np.number])  
numeric = phones\_new\_numeric.columns.values*  
*for i in numeric:  
 plt.hist(phones\_new[i])  
 plt.title(i)  
 plt.show()*

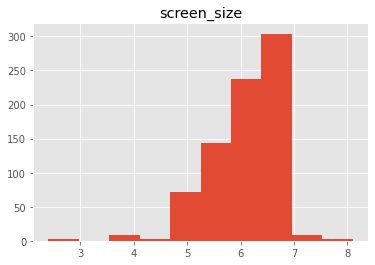


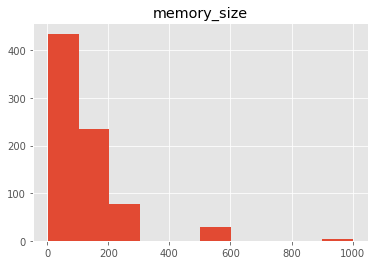


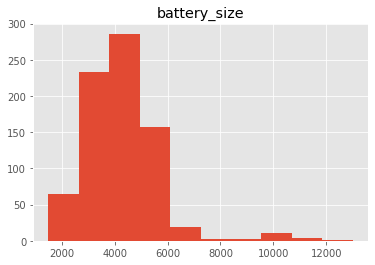






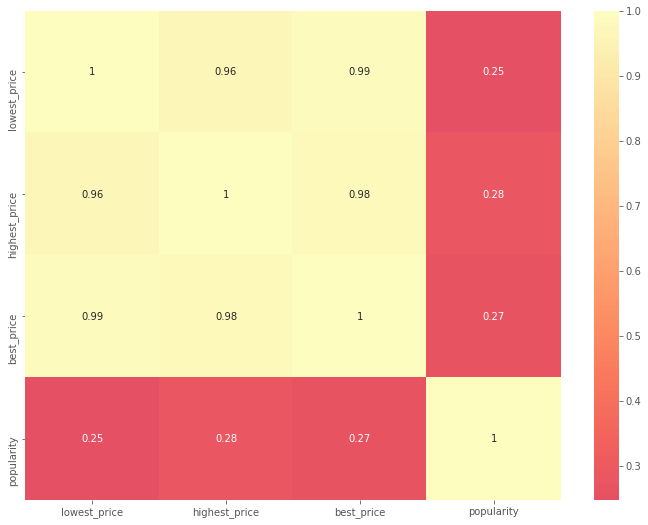






Проведемо кореляційний аналіз. Скориставшись вбудованою в pandas функцією corr(), побудуємо кореляційну матрицю числових ознак датасету:

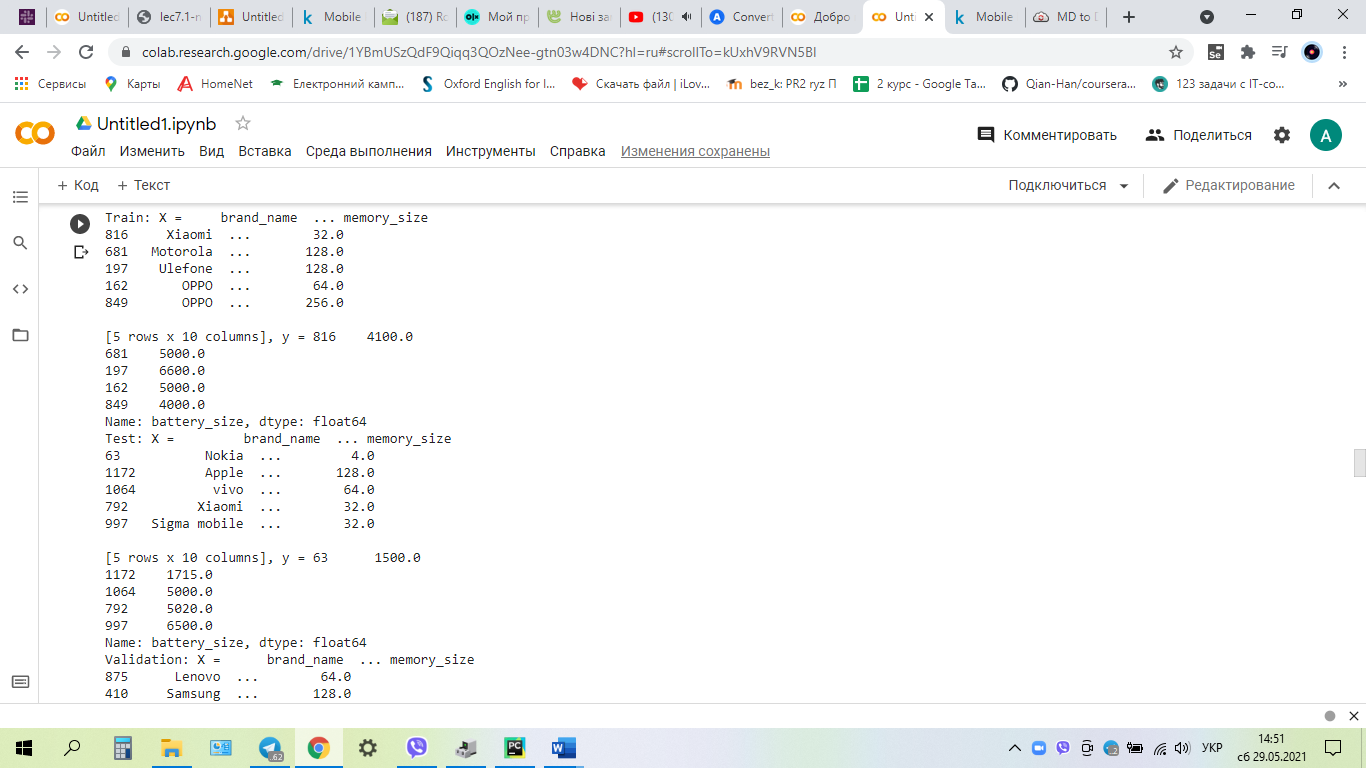
*dumm = pandas.get\_dummies(phones\_new[['lowest\_price','highest\_price','best\_price','popularity']])  
corr = pandas.DataFrame(dumm).corr()  
fig, ax = plt.subplots(1, 1, figsize=(12, 9))  
sns.heatmap(corr, cmap='magma', center=0, annot=True, ax=ax)  
plt.show()*

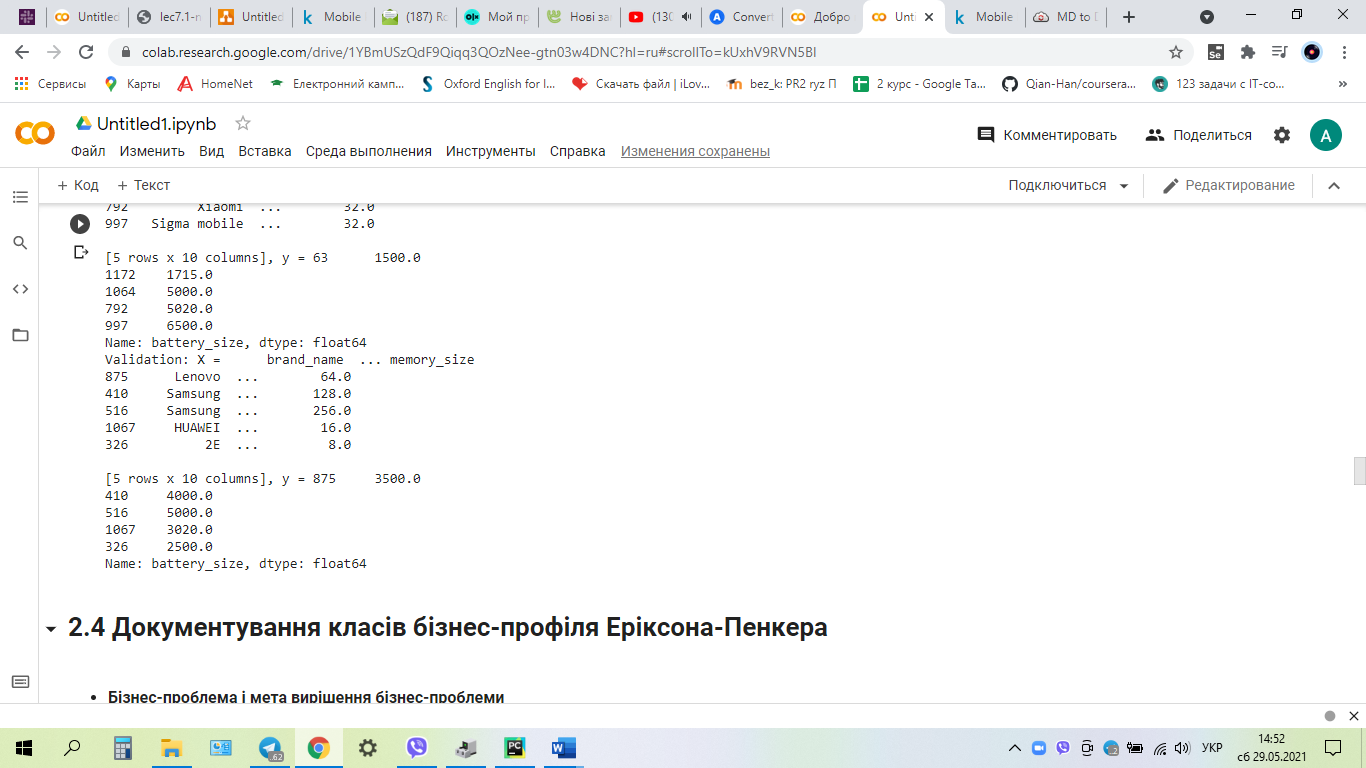


З кореляційної матриці бачимо, що взаємозв'язок між багатьма ознаками є дуже сильний. Найнижчу кореляцію мають popularity i lowest\_price(популярність та найнижча ціна). Найкраща ціна та популярність сильно корельовані між собою.

Розіб'ємо dataset на набори даних: тренувальний, валідаційний та тестовий набори. Скористаємося для цього функцією train\_test\_split().

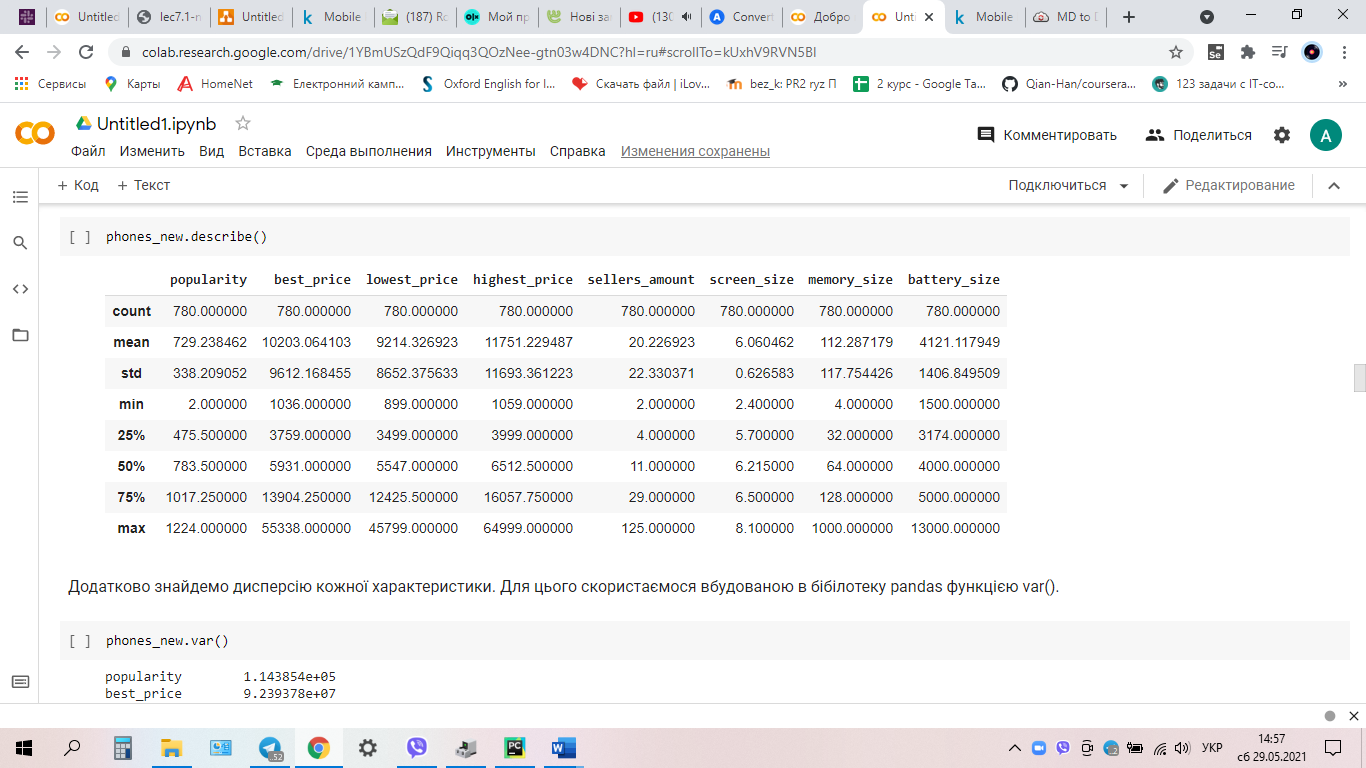
*del phones\_new['release\_date']   
X = phones\_new[phones\_new.columns[:-1]]  
y = phones\_new[phones\_new.columns[-1]]  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y)  
X\_train, X\_val, y\_train, y\_val = train\_test\_split(X\_train, y\_train)  
  
print(f'Train: X = {X\_train.head()}, y = {y\_train.head()}')  
print(f'Test: X = {X\_test.head()}, y = {y\_test.head()}')  
print(f'Validation: X = {X\_val.head()}, y = {y\_val.head()}')*





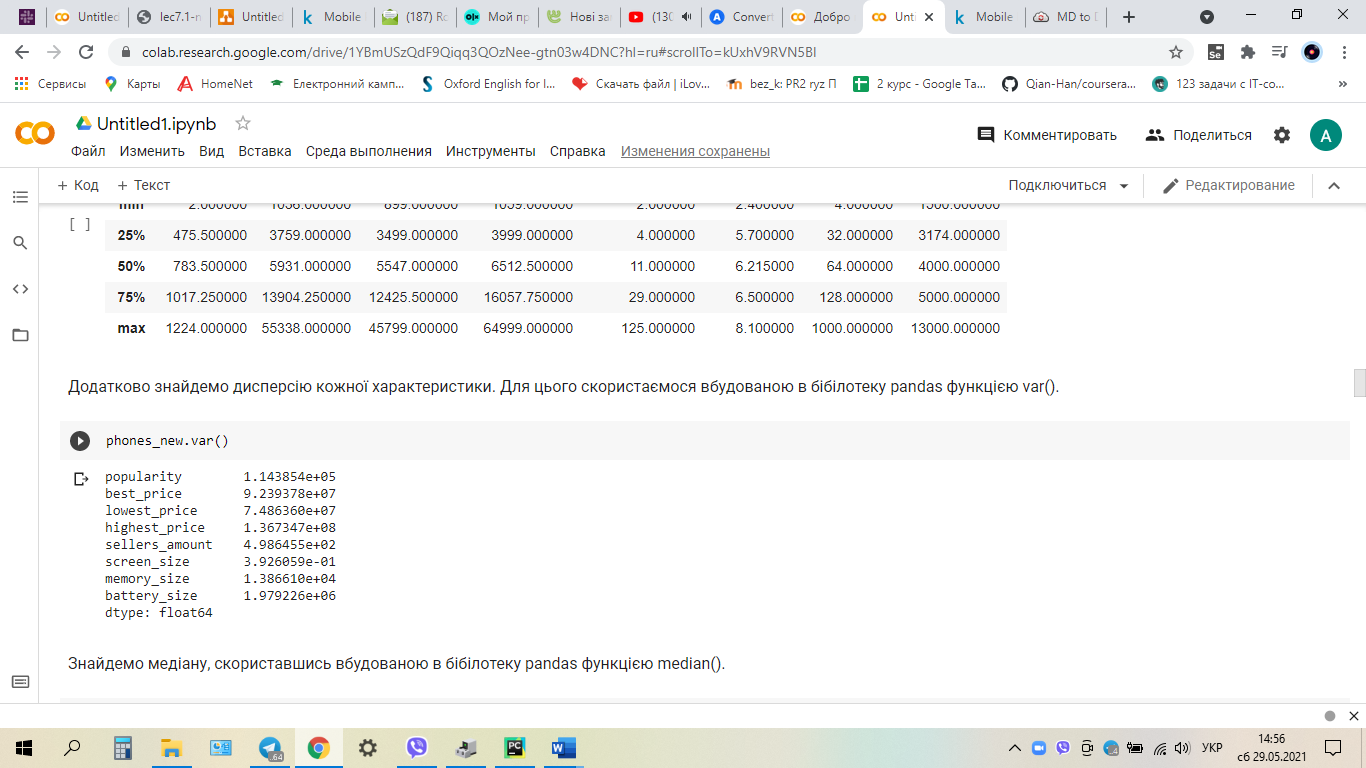
Застосуємо описову статистику: середні арифметичні, стандартні відхилення, мінімальне та максимальне значення кожної характеристики, скориставшись функцією describe():

*phones\_new.describe()*



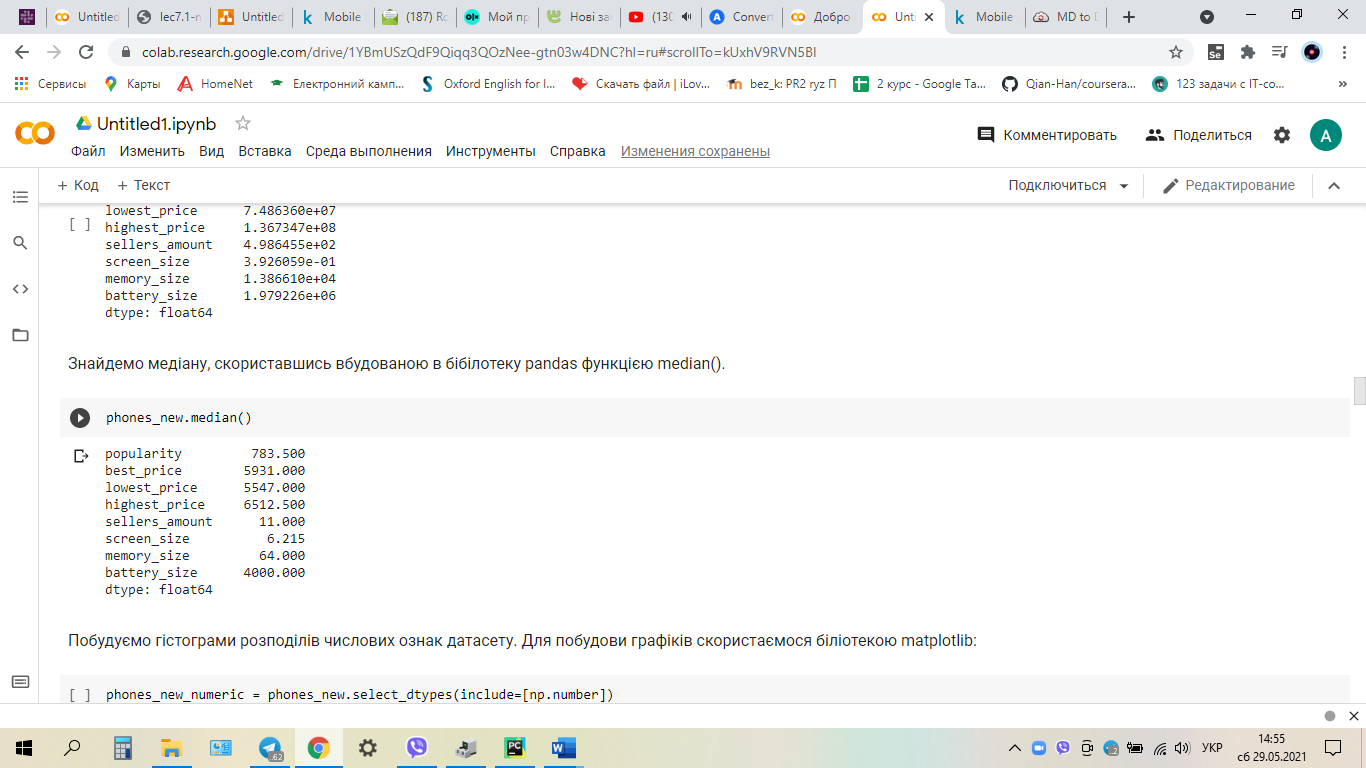
Додатково знайдемо дисперсію кожної характеристики. Для цього скористаємося вбудованою в бібілотеку pandas функцією var().

*phones\_new.var()*



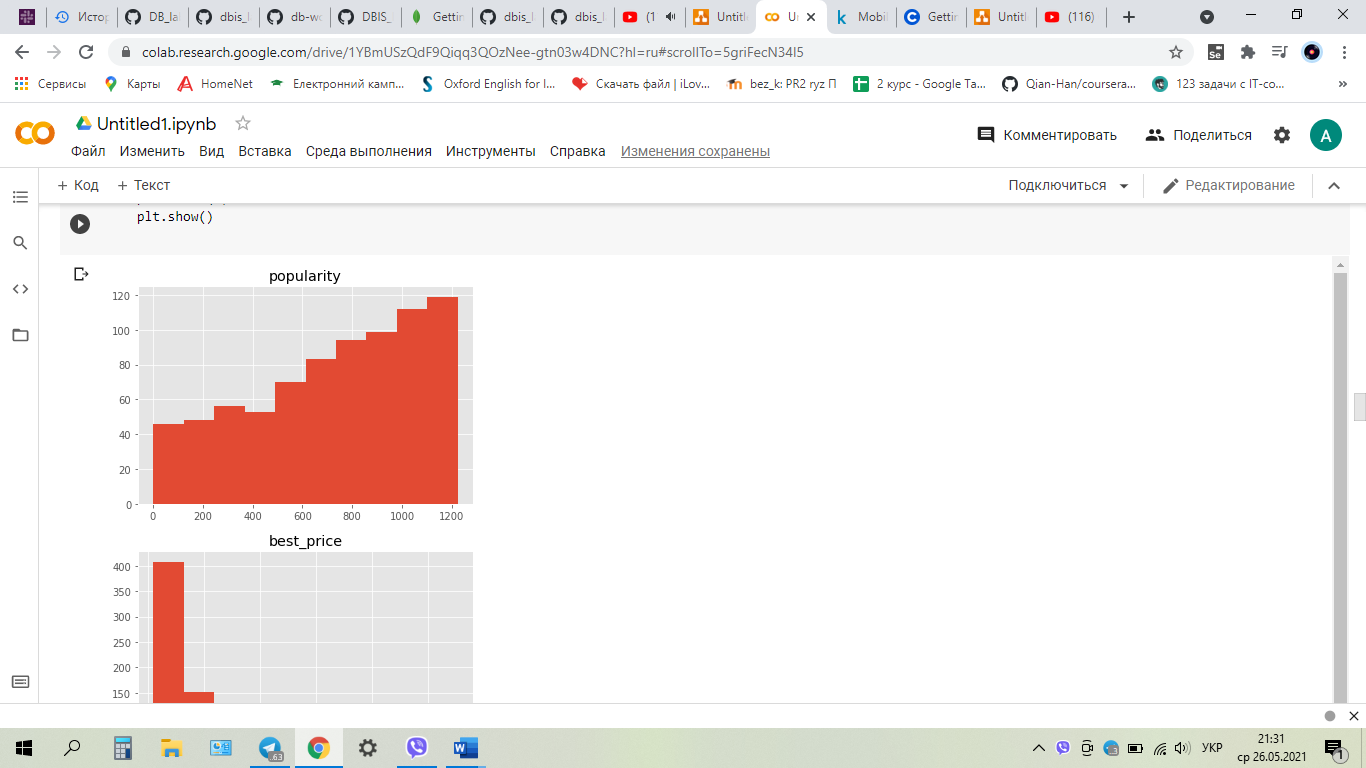
Знайдемо медіану, скориставшись вбудованою в бібілотеку pandas функцією median().

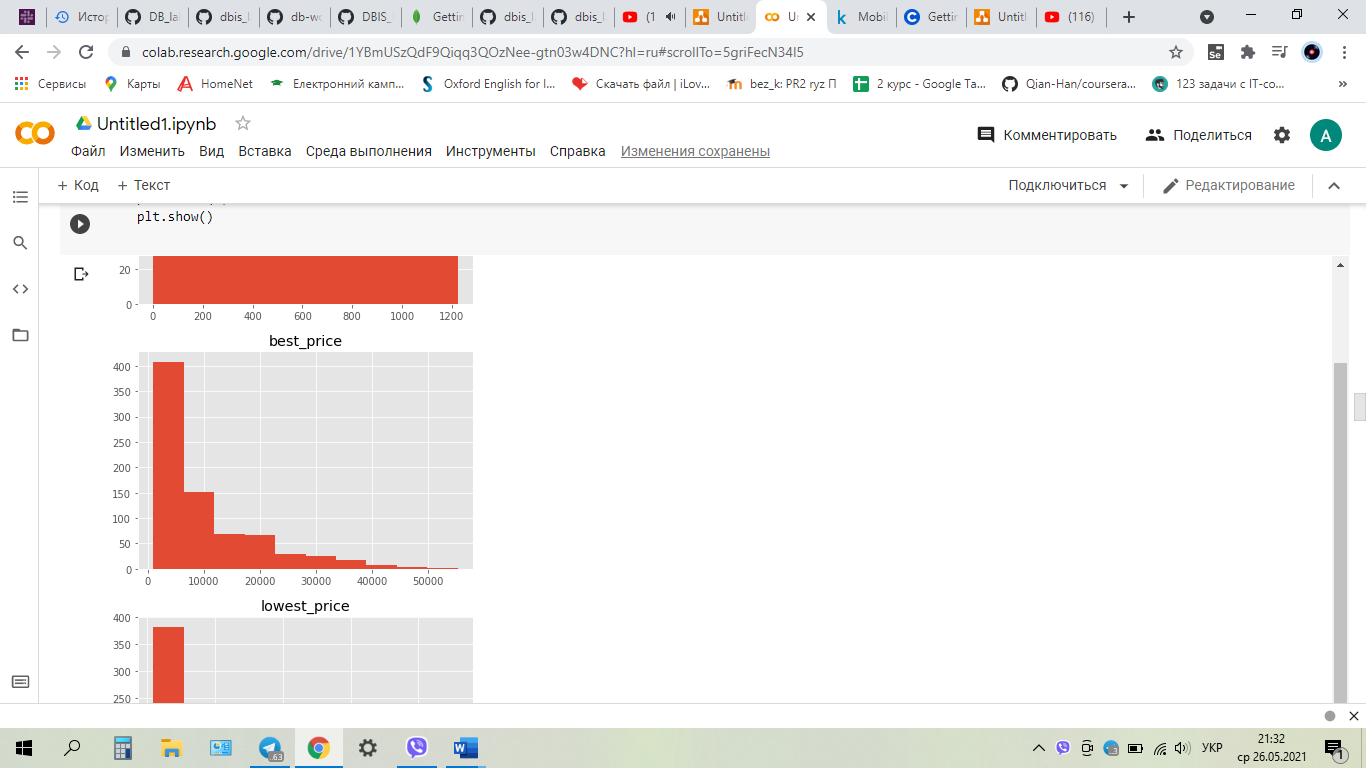
*phones\_new.median()*

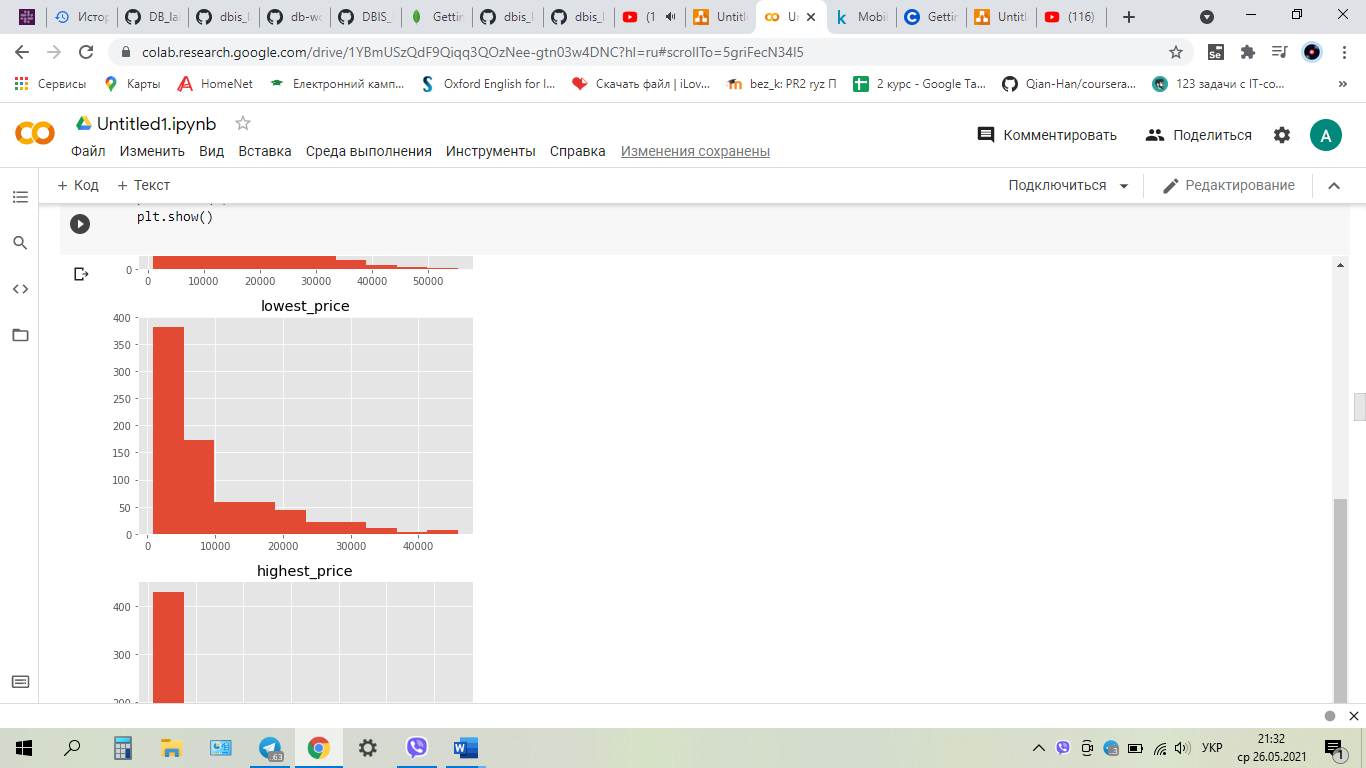


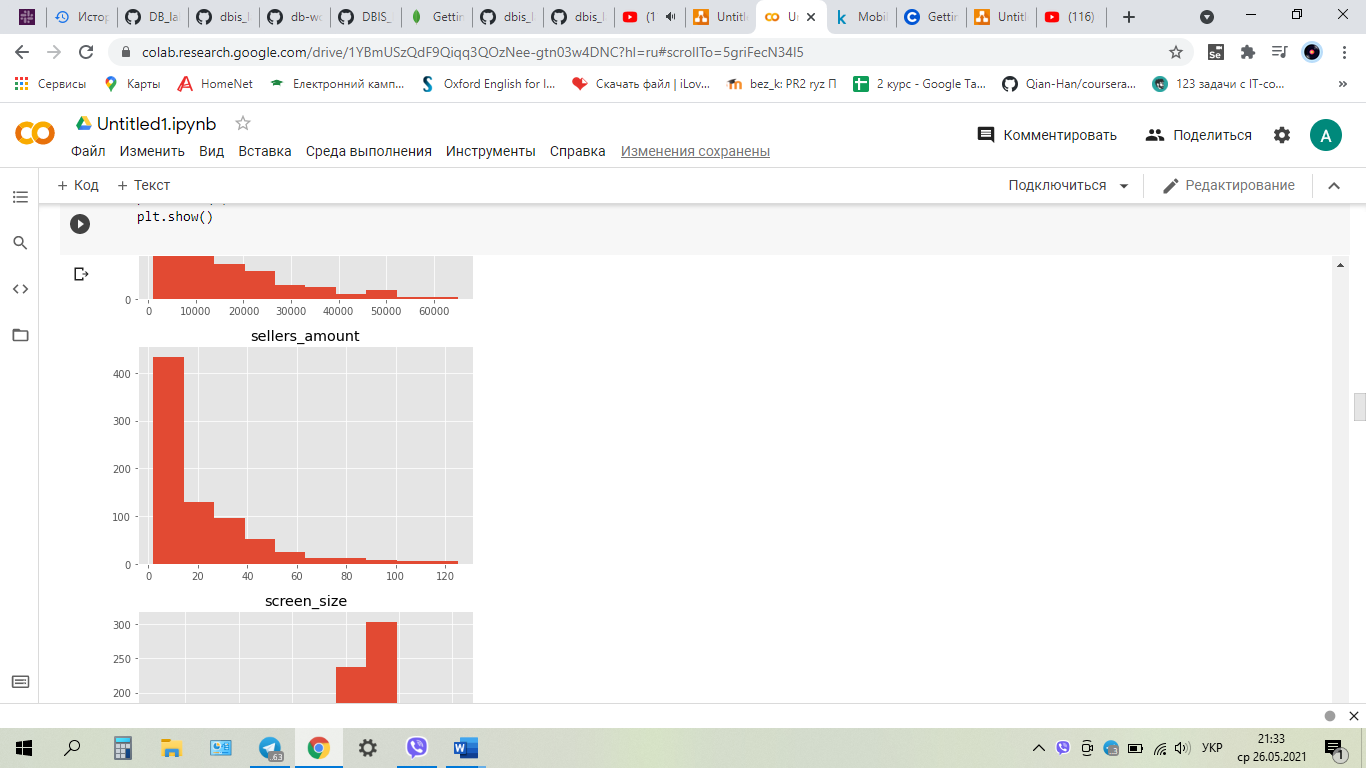
Побудуємо гістограми розподілів числових ознак датасету. Для побудови графіків скористаємося біліотекою matplotlib:

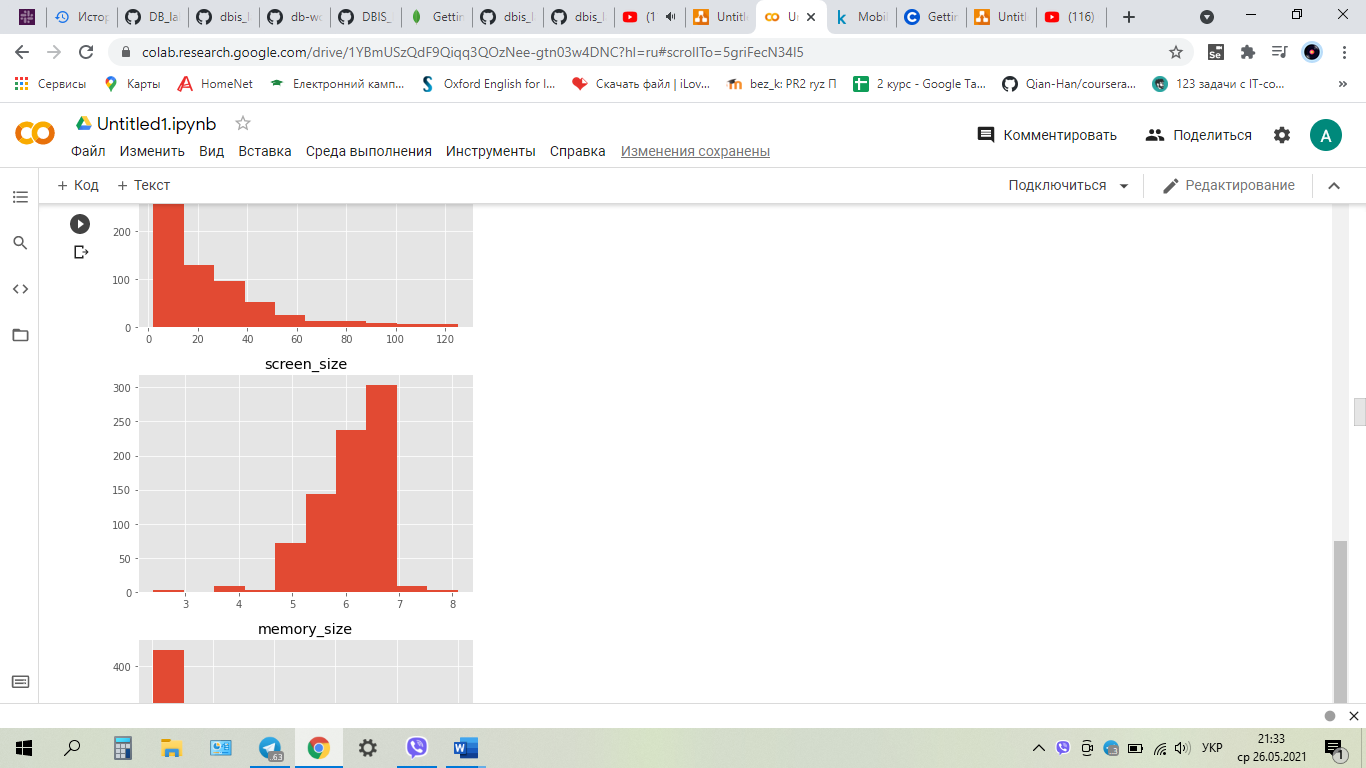
*phones\_new\_numeric = phones\_new.select\_dtypes(include=[np.number])  
numeric = phones\_new\_numeric.columns.values*  
*for i in numeric:  
 plt.hist(phones\_new[i])  
 plt.title(i)  
 plt.show()*



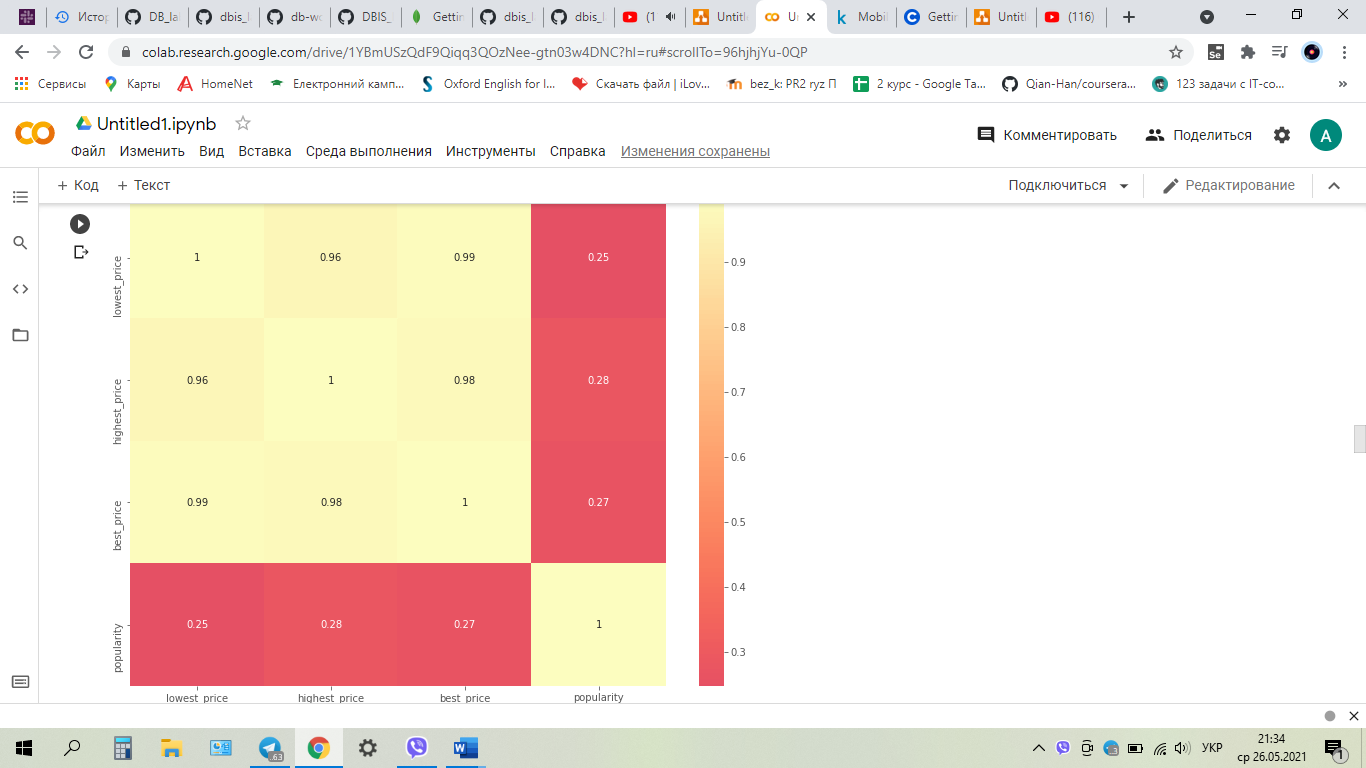








Проведемо кореляційний аналіз. Скориставшись вбудованою в pandas функцією corr(), побудуємо кореляційну матрицю числових ознак датасету:



З кореляційної матриці бачимо, що взаємозв'язок між багатьма ознаками є дуже сильний. Найнижчу кореляцію мають popularity i lowest\_price(популярність та найнижча ціна). Найкраща ціна та популярність сильно корельовані між собою.

# Формалізація цільової функції оптимізації. Визначення метрик оцінки ефективності моделі

Нехай X - матриця, стовпчики якої - це значення незалежних змінних, а рядки - це окремі мобільні пристрої.

Цільова функція оптимізації : .

Метрика оцінки ефективності моделі -- коефіцієнт детермінації . Він показує, яка частка загальної дисперсії даних пояснюється регресією.

Формула:

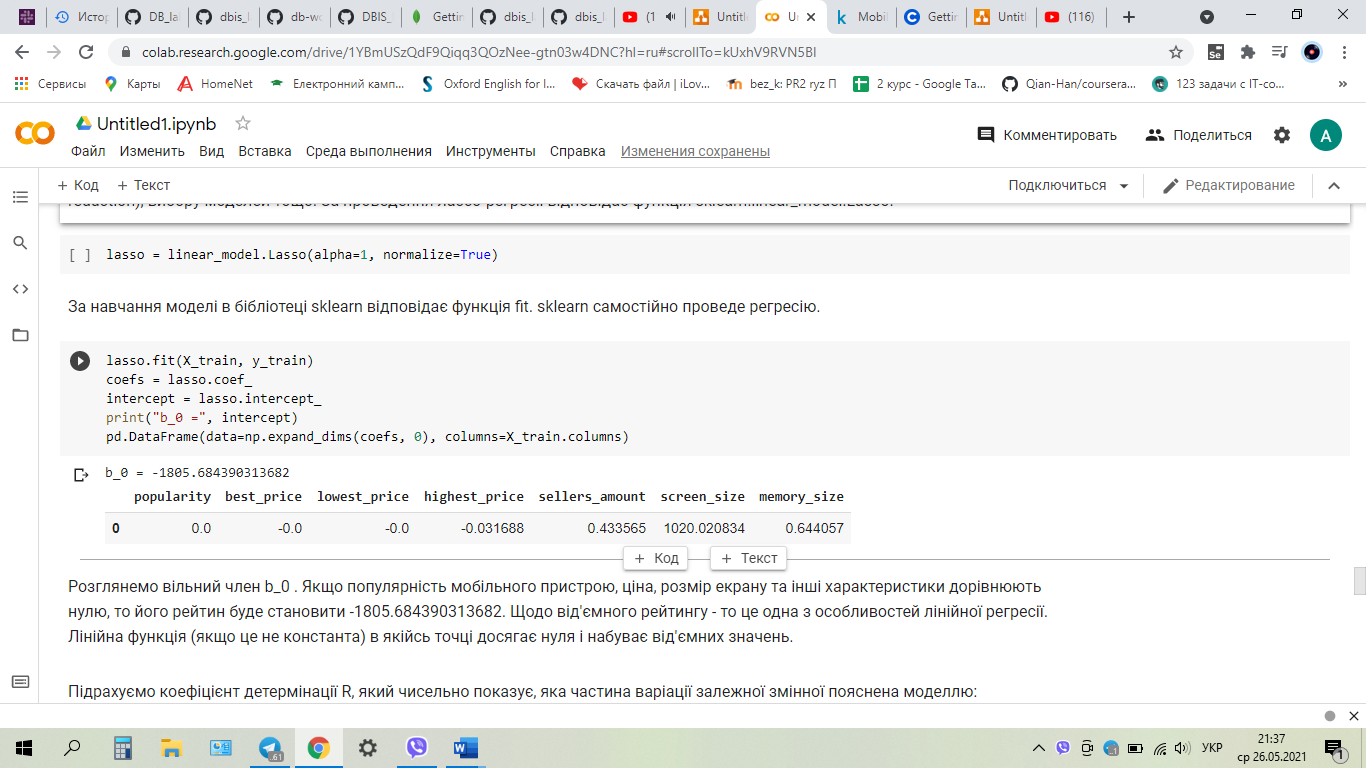
**Створення моделі**

Для задачі лассо-регресії зручно використати вбудований модуль sklearn (scikit-learn). Це бібліотека Python, яка містить засоби для різноманітних задач машинного навчання: класифікації, регресії, кластеризації, зменшення розмірності простору (dimensionality reduction), вибору моделей тощо. За проведення лассо-регресії відповідає функція sklearn.linear\_model.Lasso.

*lasso = linear\_model.Lasso(alpha=1, normalize=True)*

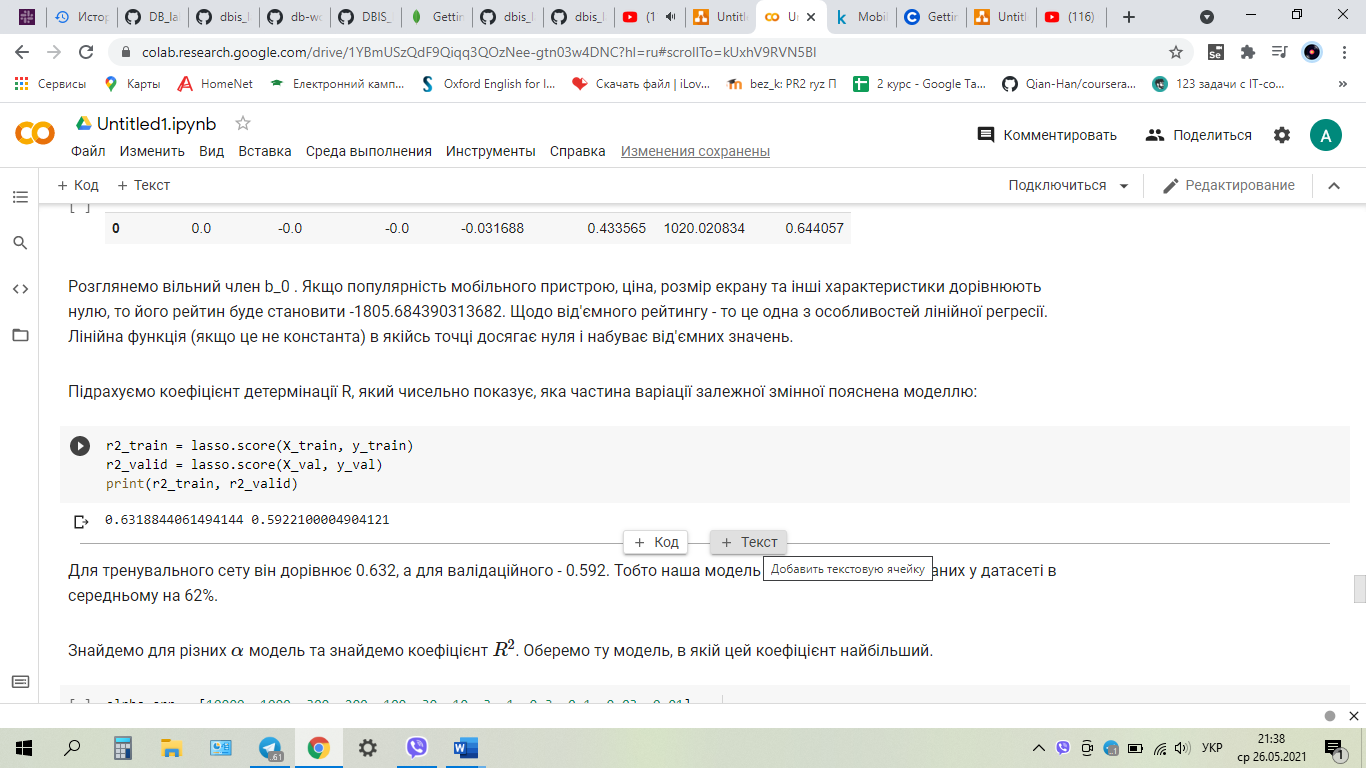
За навчання моделі в бібліотеці sklearn відповідає функція fit. sklearn самостійно проведе регресію.

*lasso.fit(X\_train, y\_train)  
coefs = lasso.coef\_  
intercept = lasso.intercept\_  
print("b\_0 =", intercept)  
pd.DataFrame(data=np.expand\_dims(coefs, 0), columns=X\_train.columns)*



Розглянемо вільний член b\_0 . Якщо популярність мобільного пристрою, ціна, розмір екрану та інші характеристики дорівнюють нулю, то його рейтин буде становити -1805.684390313682. Щодо від'ємного рейтингу - то це одна з особливостей лінійної регресії. Лінійна функція (якщо це не константа) в певній точці досягає нуля і набуває від'ємних значень.

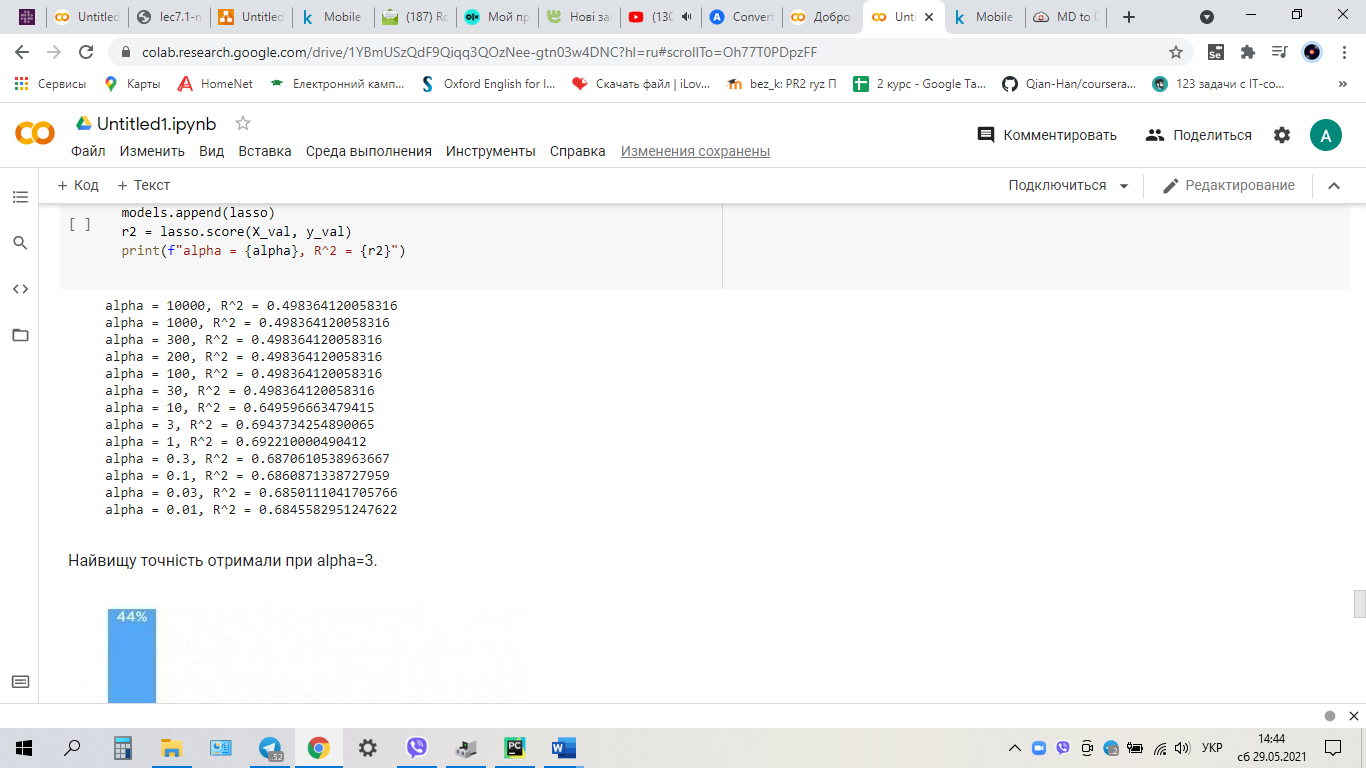
Підрахуємо коефіцієнт детермінації R, який чисельно показує, яка частина варіації залежної змінної пояснена моделлю:



Для тренувального сету він дорівнює 0.632, а для валідаційного - 0.592. Тобто наша модель пояснює мінливість даних у датасеті в середньому на 62%.

Знайдемо для різних модель та знайдемо коефіцієнт . Оберемо ту модель, в якій цей коефіцієнт найбільший.

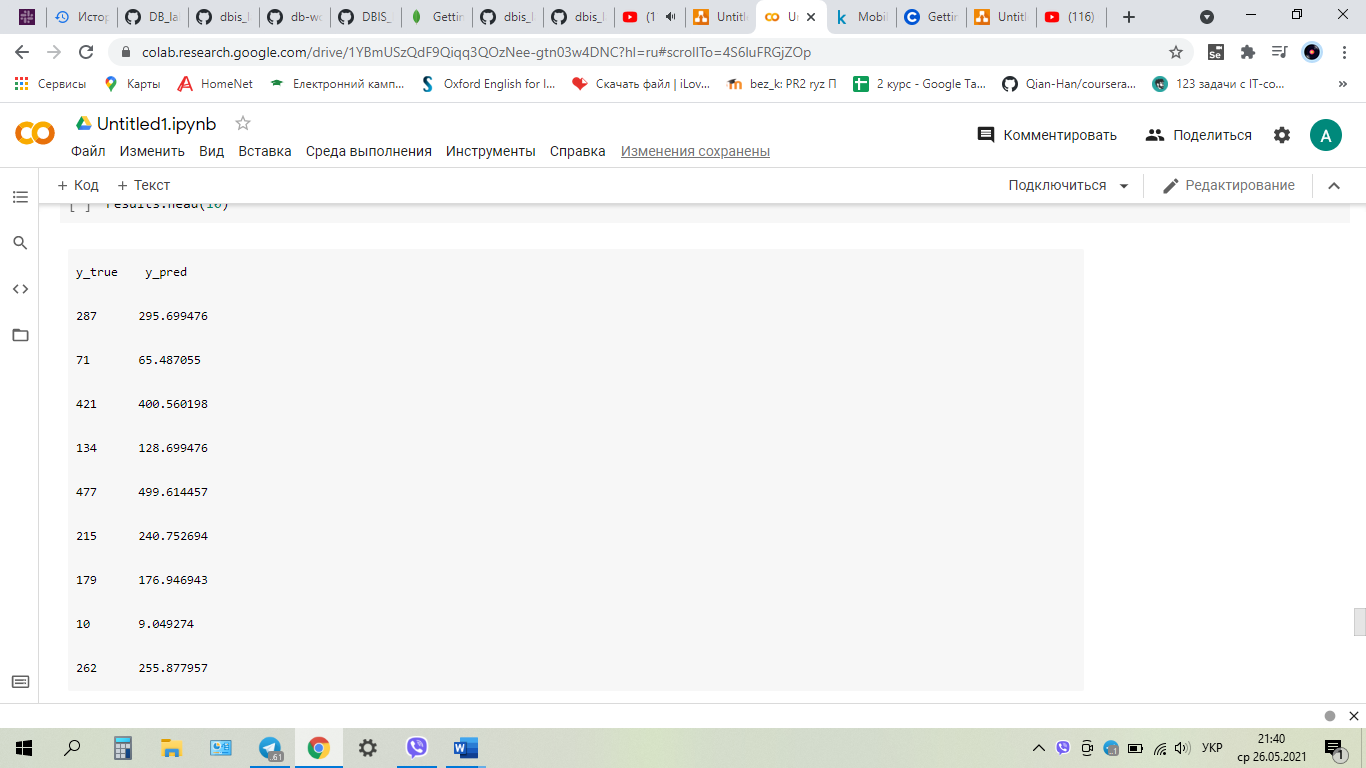
*alpha\_arr = [10000, 1000, 300, 200, 100, 30, 10, 3, 1, 0.3, 0.1, 0.03, 0.01]  
models = []  
for alpha in alpha\_arr:  
 lasso = linear\_model.Lasso(alpha=alpha, normalize=True)  
 lasso.fit(X\_train, y\_train)  
 models.append(lasso)  
 r2 = lasso.score(X\_val, y\_val)  
 print(f"alpha = {alpha}, R^2 = {r2}")*



Найвищу точність отримали при alpha=3.

![image.png](data:image/png;base64;base64,)

*results = pd.DataFrame({'y\_true': np.squeeze(Y\_test), 'y\_pred':Y\_pred})  
results.head(10)*



Бачимо, що рейтинг прогнозований нашою моделлю (друга колонка) трохи відрізняється від реального значення популярності мобільного телефону. Якщо популярність телефону складає 287, то модель передбачила 296. Можна помітити закономірність, що чим менша популярність телефону, тим точніше наша модель змогла її спрогнозувати.

# Інтерпретація отриманих результатів

В контексті задачі моделювання популярності мобільних пристроїв був обраний метод лассо-регресія. Цей алгоритм легко інтерпретувати (тобто зрозуміти, що саме намагається зробити цей алгоритм). Було виявлено які змінні майже не впливають на рейтинг телефонів, наприклад найнижча ціна.

Серед плюсів цього методу можна відмітити вирішення проблем корельованих вхідних даних.

Для підвищення точності моделі можна зібрати дані про більшу кількість мобільних пристроїв і додати нові характеристики про телефон, такі як інформація про процесор та камеру. Ці характеристики можуть позитивно вплинути на рейтинг мобільного пристрою.

**Мета вирішення задачі моделювання:** в даній роботі було обрано задачу регресії. Було взято датасет "Mobile Phones Data". Задачею моделювання в рамках конкретного контексту є моделювання системи для передбачення популярності мобільного пристрою.

**Процеси/функції вирішення проблеми:** процеси і алгоритми статистичного аналізу, які було застосовано: описова статистика (знаходження середніх арифметичних, стандартних відхилень, мінімального та максимального значень, дисперсії, моди та медіани кожної ознаки, а також побудова гістограм розподілу кожної ознаки), кореляційний аналіз (побудовано кореляційну матрицю та проаналізовано степінь взаємозв'язку).

**Ресурси для реалізації і вирішення задачі:**

інструменти та мета їх застосування: [https://www.kaggle.com/](https://www.kaggle.com/%20) (джерело даних; збережено dataset), [https://pandas.pydata.org/docs/](https://pandas.pydata.org/docs/%20) (бібліотека, за допомогою якої було подано дані у вигляді dataframe для зручності в роботі; було використано її вбудовані функції для реалізації описової статистики та кореляційного аналізу), <https://matplotlib.org/> (бібліотека, за допомогою якої виконувався побудова графіків, а саме гістограм розподілу при реалізації описової статистики).

# ВИСНОВКИ

Перед початком реалізації моделювання системи для прогнозування рейтингу мобільних пристроїв було виконано:

* первинну очистку сирих даних (видалено стовпець unnamed,  усунення NaN/Null значень, перевірено, що записи-дублікати в датасеті відсутні).
* проведено розвідувальний аналіз (побудована кореляційна матриця, з якої було з’ясовано, що найнижчу кореляцію мають popularity i lowest\_price)
* дослідження датасету із застосуванням описової статистики
* розбиття очищених та підготовлених даних на два табори (тренувальний та тестовий).

Це нам дало базу для виконання подальшого моделювання задачі, що була запланована на початку даної роботи.

Основна ціль моделі - прогнозування рейтингу мобільних телефонів, які можна купити в Україні, тобто, є задачею регресії. Було вибрано метод лассо-регресія та на його основі побудовано модель.

Оцінено точність моделі: на тестовому датасеті дана модель пояснює 62% усієї мінливості даних. Для побудови моделі ми використали бібліотеку для мови Python - sklearn.

Система прогнозування показала, що популярними телефонами, які продаються в Україні, є Apple та Samsung. Ця інформація відповідає дійсності.



Дана система може бути корисною для корпорацій-виробників мобільних телефонів, які бажають дізнатися чи буде користуватися попитом та чи інша модель телефону, а також для покупців, які зможуть орієнтуватися в рейтингу телефонів, що є одним з головних показників при покупці гаджетів.

Отже, поставлена задача виконана, а результат, отриманий по завершенню виконання задачі цілком влаштовує цілі, поставлені перед початком моделювання.

*import pandas*

*from sklearn.model\_selection import train\_test\_split*

*import seaborn as sns*

*import matplotlib.pyplot as plt*

*import matplotlib.mlab as mlab*

*import matplotlib*

*import numpy as np*

*plt.style.use('ggplot')*

*from matplotlib.pyplot import figure*

*phones = 'drive/MyDrive/phones\_data.csv'*

*phones\_new = phones.drop(columns="Unnamed: 0")*

*phones\_new*

*phones\_new.isnull().sum()*

*phones\_new.dropna(subset=['os','lowest\_price', 'highest\_price', 'screen\_size', 'memory\_size', 'battery\_size'], inplace=True)  
phones\_new*

*phones\_new.isnull().sum()*

*phones\_new.describe()*

*phones\_new.var()*

*phones\_new.median()*

*phones\_new\_numeric = phones\_new.select\_dtypes(include=[np.number])  
numeric = phones\_new\_numeric.columns.values*  
*for i in numeric:  
 plt.hist(phones\_new[i])  
 plt.title(i)  
 plt.show()*

dumm **=** pandas**.**get\_dummies(phones\_new[['lowest\_price','highest\_price','best\_price','popularity']])

In [22]:

corr **=** pandas**.**DataFrame(dumm)**.**corr()

fig, ax **=** plt**.**subplots(1, 1, figsize**=**(12, 9))

sns**.**heatmap(corr, cmap**=**'magma', center**=**0, annot**=True**, ax**=**ax)

plt**.**show()

*lasso = linear\_model.Lasso(alpha=1, normalize=True)*

*lasso.fit(X\_train, y\_train)  
coefs = lasso.coef\_  
intercept = lasso.intercept\_  
print("b\_0 =", intercept)  
pd.DataFrame(data=np.expand\_dims(coefs, 0), columns=X\_train.columns)*

*alpha\_arr = [10000, 1000, 300, 200, 100, 30, 10, 3, 1, 0.3, 0.1, 0.03, 0.01]  
models = []  
for alpha in alpha\_arr:  
 lasso = linear\_model.Lasso(alpha=alpha, normalize=True)  
 lasso.fit(X\_train, y\_train)  
 models.append(lasso)  
 r2 = lasso.score(X\_val, y\_val)  
 print(f"alpha = {alpha}, R^2 = {r2}")*

*results = pd.DataFrame({'y\_true': np.squeeze(Y\_test), 'y\_pred':Y\_pred})  
results.head(10)*