

**Full[*almost*] Data Science Project from scratch to end**

**Phoner:** Phone price predictor and not only  
FromIdea2Idea

\$~: whoami

Aziz Nadirov

Data Scientist|MLOps Engineer@Umico



<https://www.linkedin.com/in/aziz-nadirov/>



[https://www.youtube.com/@lt\\_academy](https://www.youtube.com/@lt_academy)  
(self promotion)

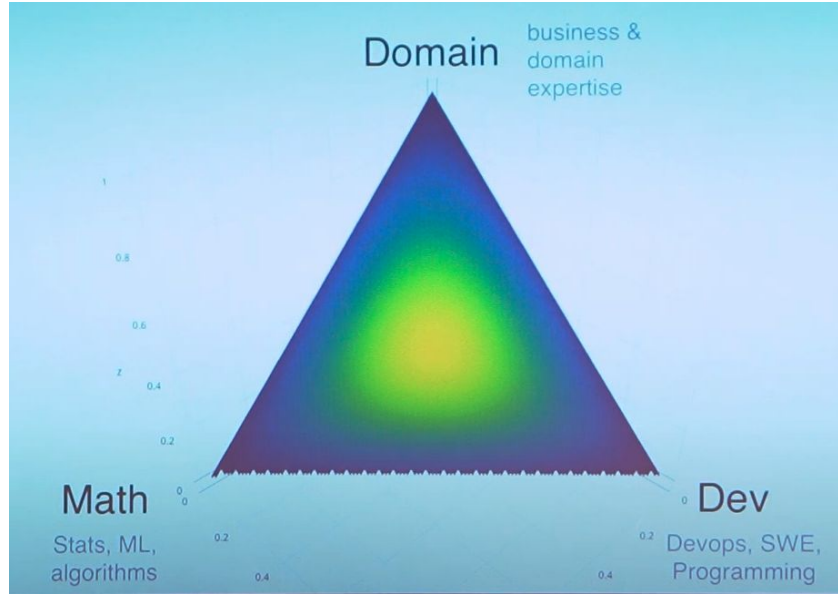


<https://github.com/AzizNadirov>

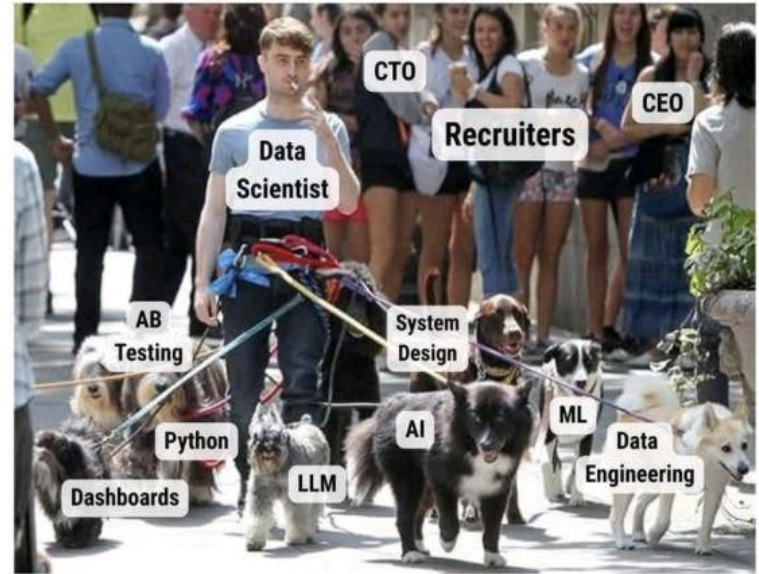


@DS\_CLUB\_BAKU

## Data Science nowadays:

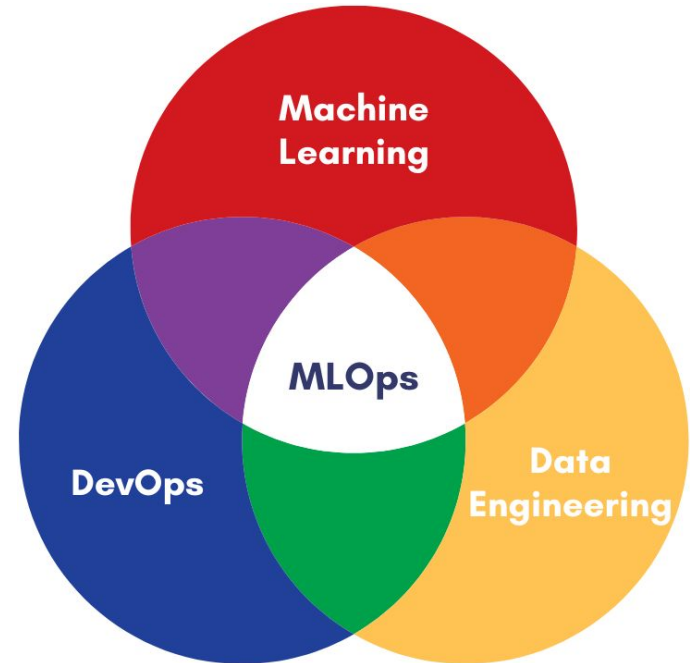


## Being a Data Scientist in 2025...



## Data Science today := Computes Science 90s

- ML Engineer
- Data Engineer
- Data Analytics
- MLOps Engineer
- NLP Enginner
- CV Enginner
- AI Engineer
- LLMOps Engineer
- ...
- Prompt Engineer :D



**Today's Mission:** show real-world workflow: from idea to  
- implementation[?]

“””

*İstifadəçilərə “necəsə” telefon qiymətlərində məsləhət verəcək app yaratmaq.  
Telefonun qiymətini təxmin edilən qiymətlə müqayisə edib, ən yaxşı seçim etməyə  
kömək olmalıdır.*

“””

# Təqribi Texniki Stek

“”””

*İstifadəçilərə “necəsə” telefon qiymətlərində məsləhət verəcək app yaratmaq.  
Telefonun qiymətini təxmin edilən qiymətlə müqayisə edib, ən yaxşı seçim  
etməyə kömək olmalıdır.*

“”””

1. Söhbət qiymət təxmin etməkdən gedirdi - ML model [?]
2. Telefonu götürüb qiyməti təxmin etmək lazımdı - parsing lazım olacaq.
3. Götürdüyümüz telefonu modelə verməliyik, deməli pars olunanları modelə uyğun hala salmaq lazımdır - matching məsələsi.
4. Nəticəni istifadəçi-yönümlü vermək lazımdır - bir az analitika lazım olacaq.
5. Sonda: bütün bunlar web App içərisində olacaq [ sadəlik üçün streamlit ]

# 1. ML: Dataset

```
3 data = pd.read_parquet("/home/tengo/Documents/code/phoner/src/data/phone-price.parquet")
4
5 print("Shape: ", data.shape)
6 print("Columns: ", data.columns.to_list())
7
```


✓ 0.4s

Python



Shape: (165592, 11)

Columns: ['Prices', 'product\_name', 'ROM\_MB', 'RAM\_MB', 'NFC', 'camera\_mp\_float', 'CPU', 'OS', 'brand', 'model', 'Prices\_avg']

1 data

✓ 0.0s  Open 'data' in Data Wrangler

Python

	# Prices	 product_name	# ROM_MB	# RAM_MB	 NFC	# camera_mp_flo
0	229.99	Samsung Galaxy A7 3GB/32GB Gold	32768.0	3072.0	True	
1	1209.99	Samsung Galaxy Note 8 6GB/64 GB	65536.0	6144.0	True	
2	499.0	Samsung Galaxy Note 8 6GB/64 GB	65536.0	6144.0	True	
3	119.99	ZTE Blade V8 Mini 3GB/32GB Gold	32768.0	3072.0	Missing value	
4	385.0	Xiaomi Mi A1 4GB/64GB Gold	65536.0	4096.0	False	
5	389.0	Xiaomi Mi A1 4GB/64GB Gold	65536.0	4096.0	False	
6	394.9	Xiaomi Mi A1 4GB/64GB Gold	65536.0	4096.0	False	
7	399.0	Xiaomi Mi A1 4GB/64GB Gold	65536.0	4096.0	False	
8	399.99	Xiaomi Mi A1 4GB/64GB Gold	65536.0	4096.0	False	
9	279.9	Xiaomi Mi A1 4GB/64GB Gold	65536.0	4096.0	False	

165,592 rows x 11 cols 10 ▾ per page

⏪ < Page 1 of 16560 > ⏩

   ...

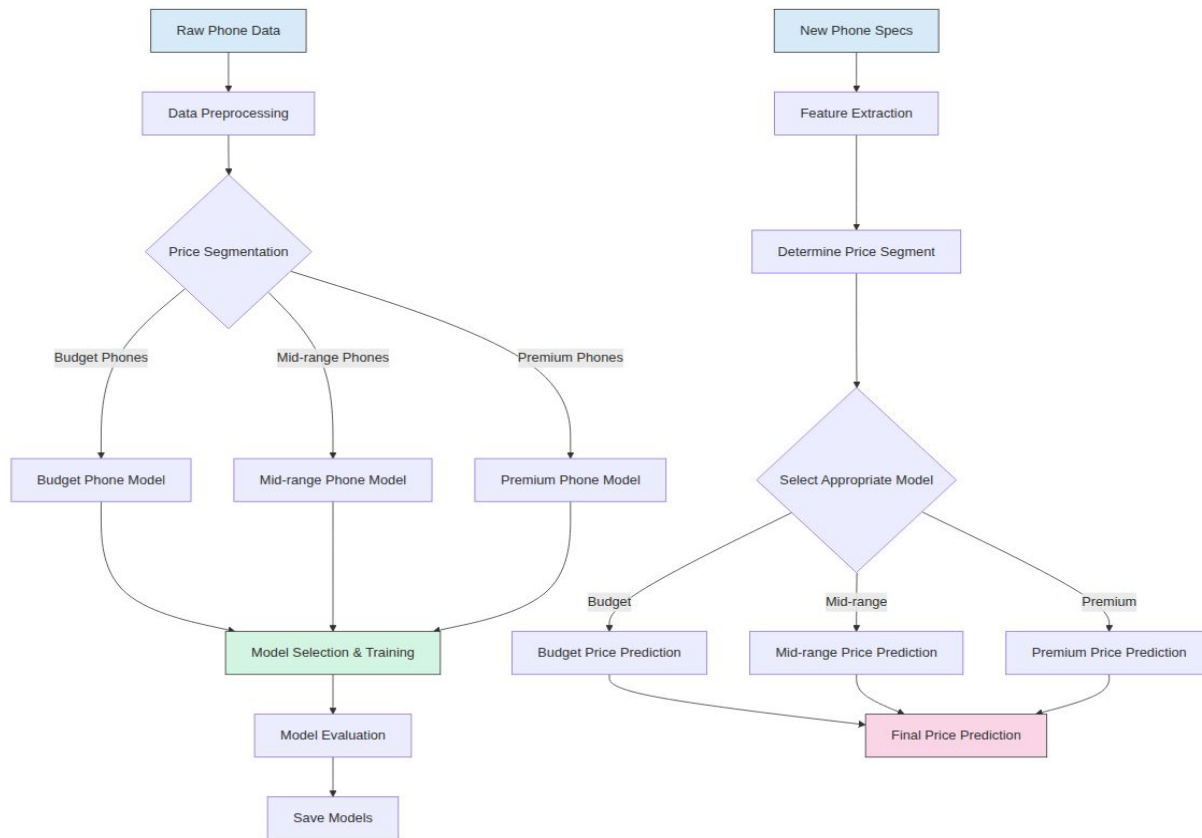
# 1. Non-ML Data Ops



# DuckDB

```
8
9 class DataSetBD:
10     """
11     duckdb conn abstraction on input dataset
12     use `.table` accessor for direct access or special methods.
13     """
14     instance = None
15
16 > def __new__(cls): ...
20
21 > def sync_from_file(self)->bool: ...
31
32 def __init__(self):
33     self.path = Path(ML_PHONE_DATASET_PATH).resolve()
34     assert os.path.exists(self.path), f"file {self.path} does not exist"
35     self.table = duckdb.query(f"SELECT * FROM '{self.path}'")
36     logger.success(f"DataSetDB Initialized.")
37
38
39 def get_avg_price_model(self,
40     | model: str) -> float | None:
41     """ returns average price for given phone 'model' """
42     logger.debug(f"Try to get model: {model}")
43     tb = 'ds'
44     query = f"""
45     select AVG(Prices_avg)
46     from {tb}
47     where model='{model}'
48     """
49     model_price = self.table.query(virtual_table_name=tb,
50     | sql_query=query).to_df()['avg(Prices_avg)'].dropna()
51
52     return model_price.values[0] if not model_price.empty else None
53
```





## 2. Crawling: crawl4ai



github.com/unclecode/crawl4ai

 **Crawl4AI: Open-Source LLM-Friendly Web Crawler & Scraper**

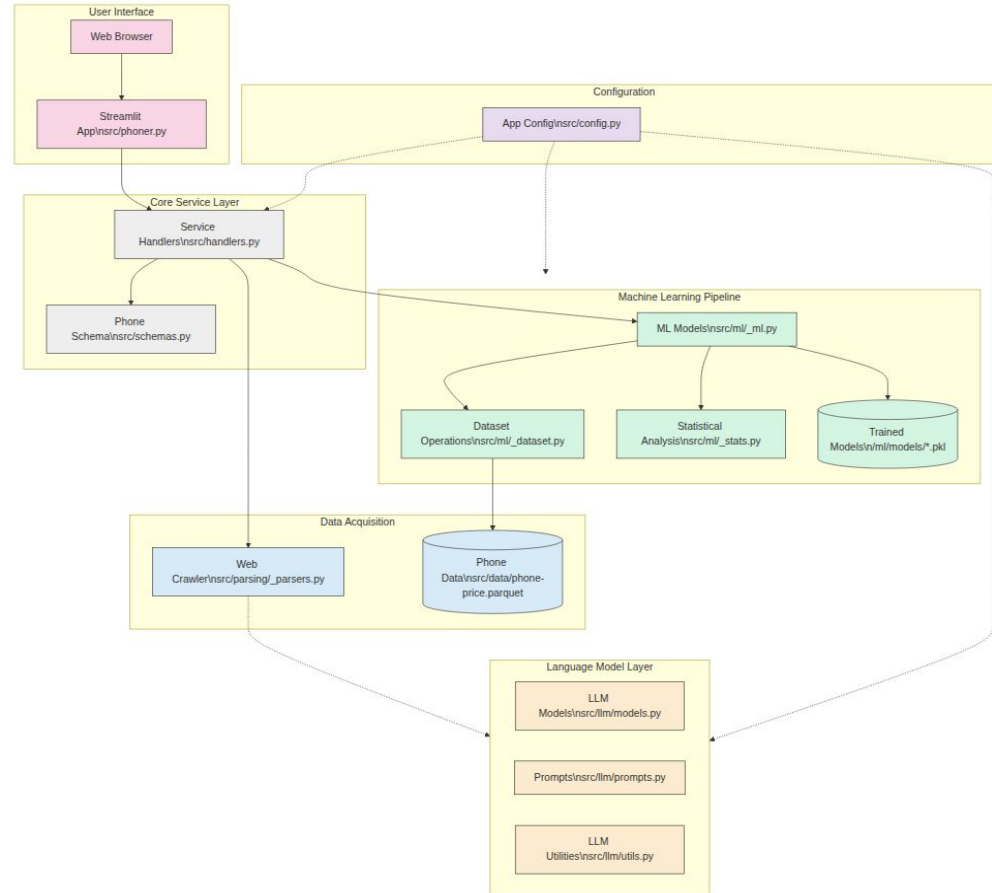
**1** GITHUB TRENDING  
**#1 Repository Of The Day**

 Stars **45k**  Forks **4.2k**  pypi package **0.6.3**

 python **3.9 | 3.10 | 3.11 | 3.12 | 3.13**  downloads/month **472k**  license **Apache-2.0**

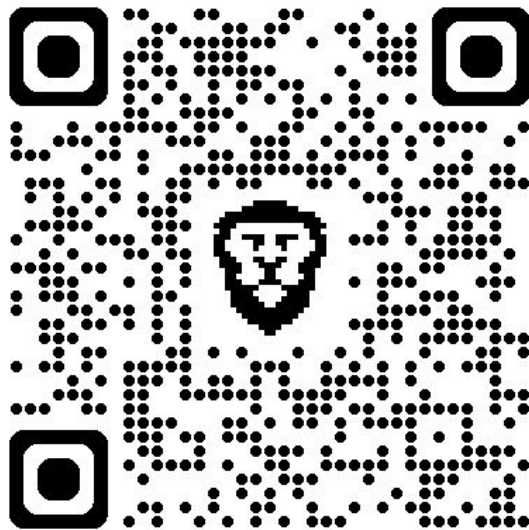


# System Design (primary)



## Ideas:

- Build phone - price - date db;
- Show price dynamics;
- Maybe add customer reviews(starts etc)
- Add non phone handling



<https://github.com/AzizNadirov/lta-phone-price-app>



**THANK YOU FOR YOUR ATTENTION**  
**now you can clap**