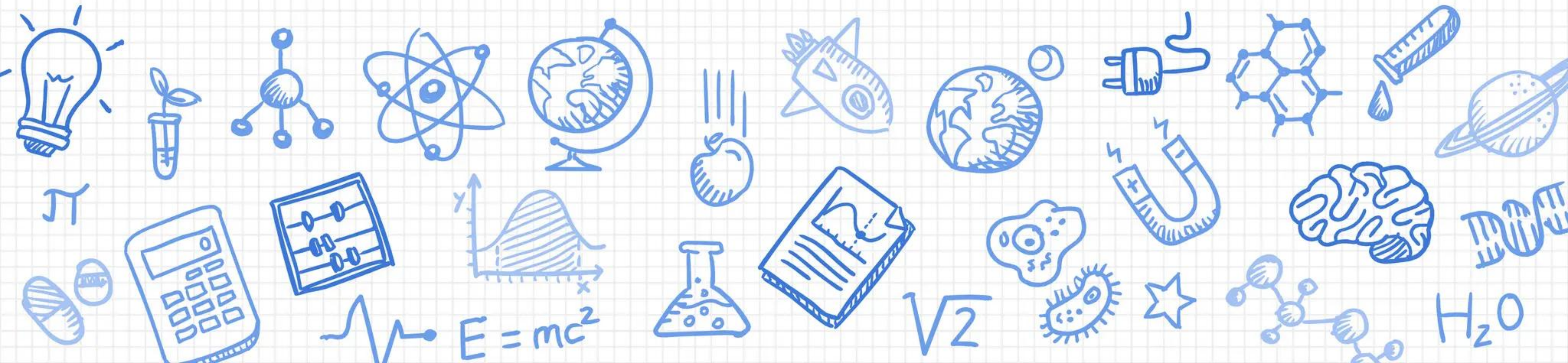


# - Data - Science and Engineering





# HELLO!

## I am Hendri Karisma

Principal R&D and Data Lead at Akar Inti Data

- ✘ Research and Development for Advanced Data Analytics Platform.
- ✘ Data marketplace at nusadata.ai .

You can find me at :

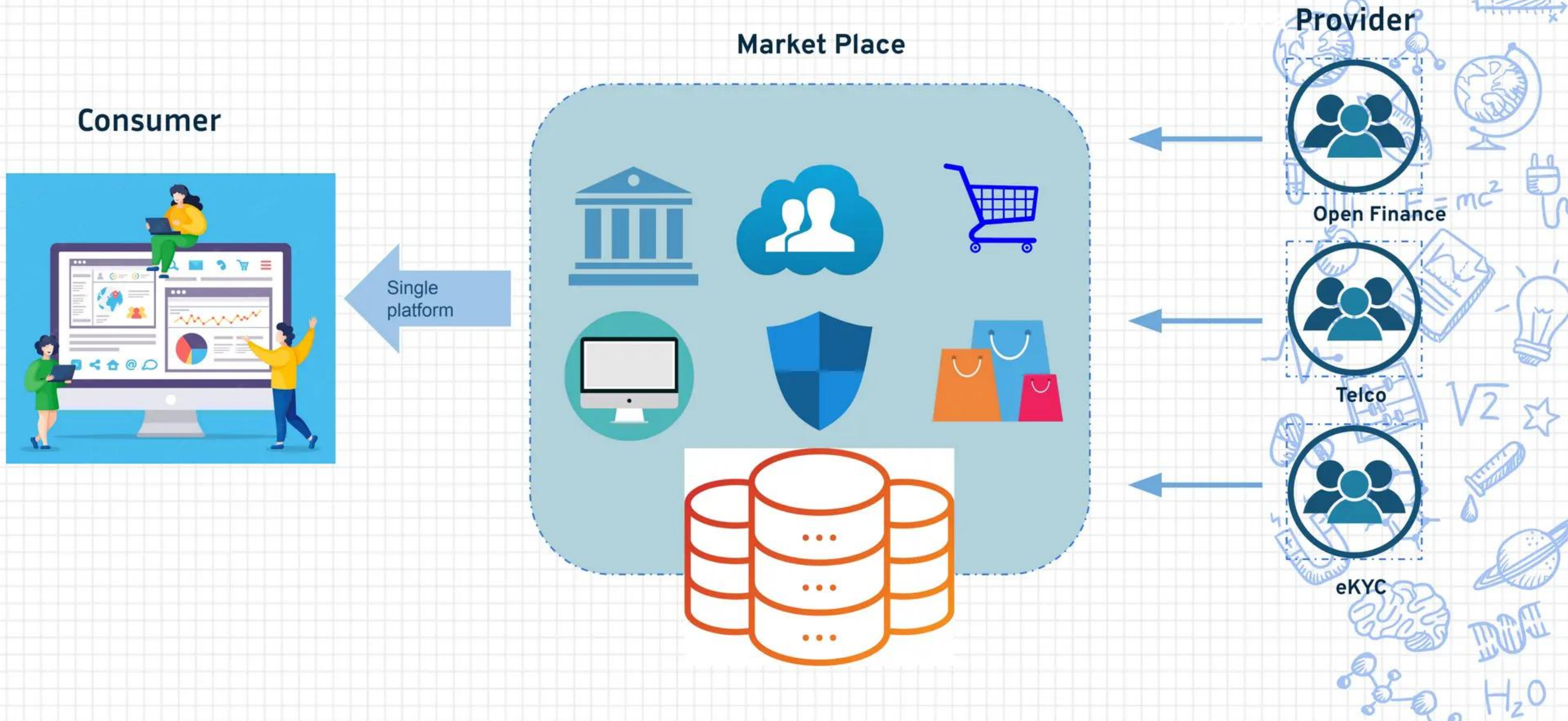
- ✘ telegram @siganteng
- ✘ Instagram @karism4\_
- ✘ X @infoHendri



# Data are Facts



# Data marketplace



# 3 Important Statistics About How Much Data Is Created Every Day

## 1 How much data is generated every minute?

Source: Domo

**41,666,667**

messages shared by WhatsApp users

**1,388,889**

video / voice calls made by people worldwide

**404,444**

hours of video streamed by Netflix users

**347,222**

stories posted by Instagram users

**150,000**

messages shared by Facebook users

**147,000**

photos shared by Facebook users

## 2 Estimated Data Consumption from 2021 to 2024

Source: IDC / Statista



## 3 Data Growth in 2021

Sources: TechJury, Internet Live Stats, Cisco, PurpleSec

**2 TRILLION**

searches on Google by the end of 2021

**1.134 TRILLION MB**

volume of data created every day

**3,026,626**

emails sent every second, 67% of which are spam

**278,108 PETABYTES**

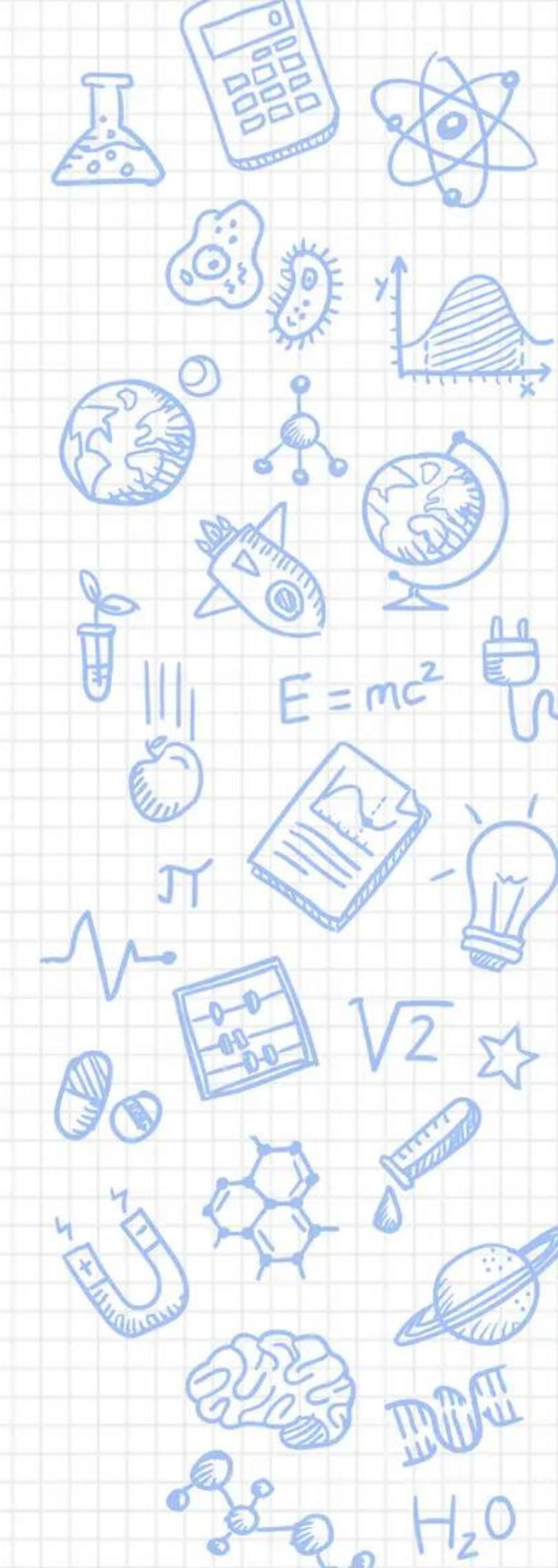
global IP data per month by the end of 2021

**230,000**

new malware versions created every day

**82%**

share of video in total global internet traffic at the end of 2021



# Science vs Engineering

## What is the difference?

---

### Science

The goal of a scientist is to answer questions and discover information about their chosen field of study.

### Engineering

An engineer might produce physical item or blueprint for a new process.

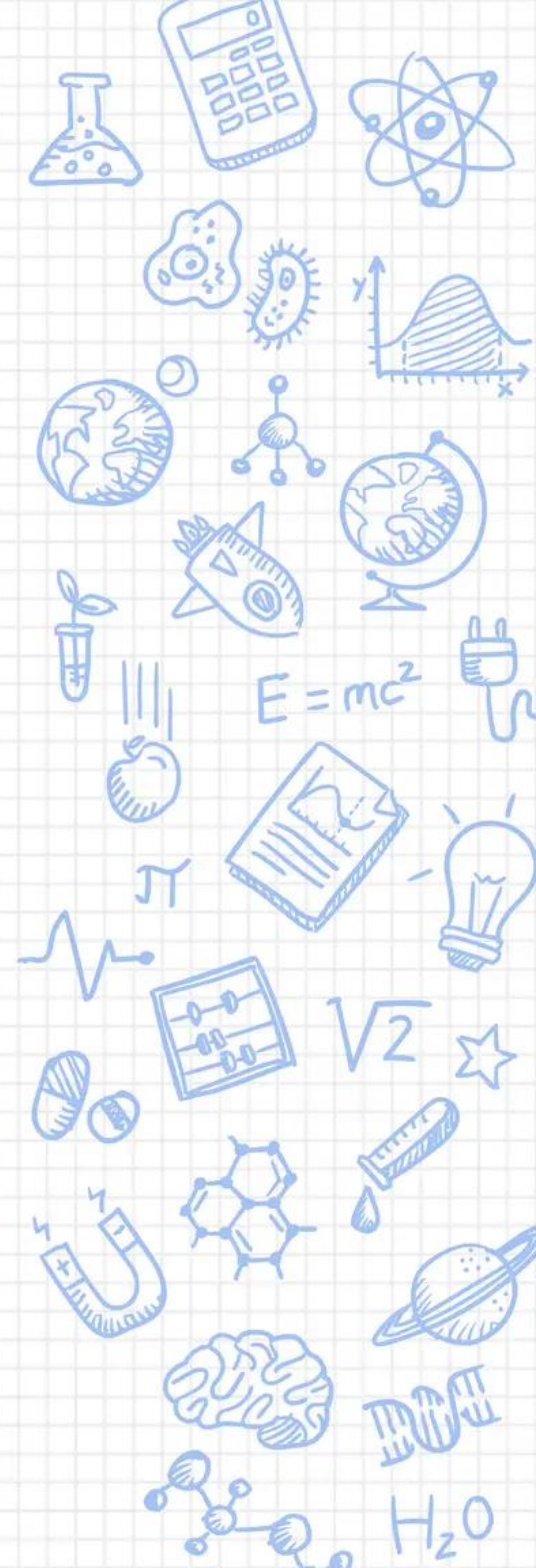


## Data roles

---

- ✘ Data **Scientist**
- ✘ Data Analyst
- ✘ Business Intelligence
- ✘ Data **Engineer**
- ✘ AI/ML **Engineer**

There is a Scientist role and 2 engineer roles.

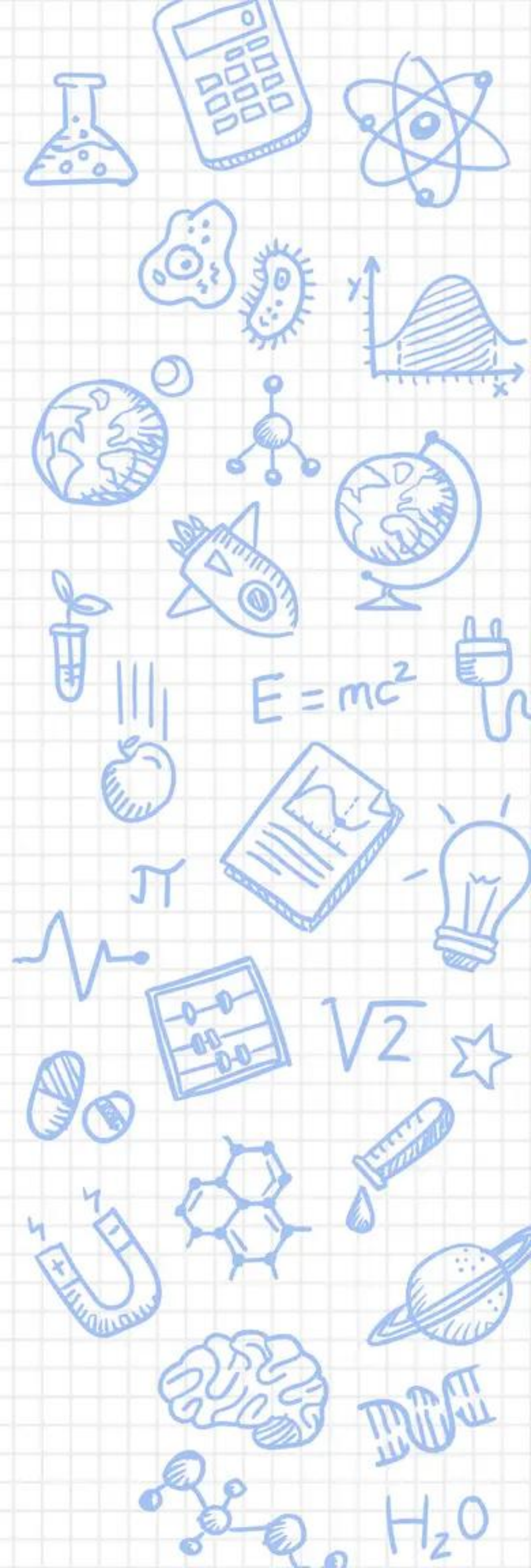


## Data Scientist

---

The term “data scientist” was coined as recently as 2008 when companies realized the need for data professionals who are skilled in organizing and analyzing massive amounts of data.<sup>1</sup> In a 2009 McKinsey&Company article, Hal Varian, Google’s chief economist and UC Berkeley professor of information sciences, business, and economics, predicted the importance of adapting to technology’s influence and reconfiguration of different industries.

Data-driven individuals with high-level technical skills who are **capable of building complex quantitative algorithms** to organize and synthesize large amounts of information used to answer questions and drive strategy in their organization.



## Data Engineer

---

Data engineers work in a variety of settings to build systems that collect, manage, and convert raw data into usable information for data scientists and business analysts to interpret. Their ultimate goal is to make data accessible so that organizations can use it to evaluate and optimize their performance.

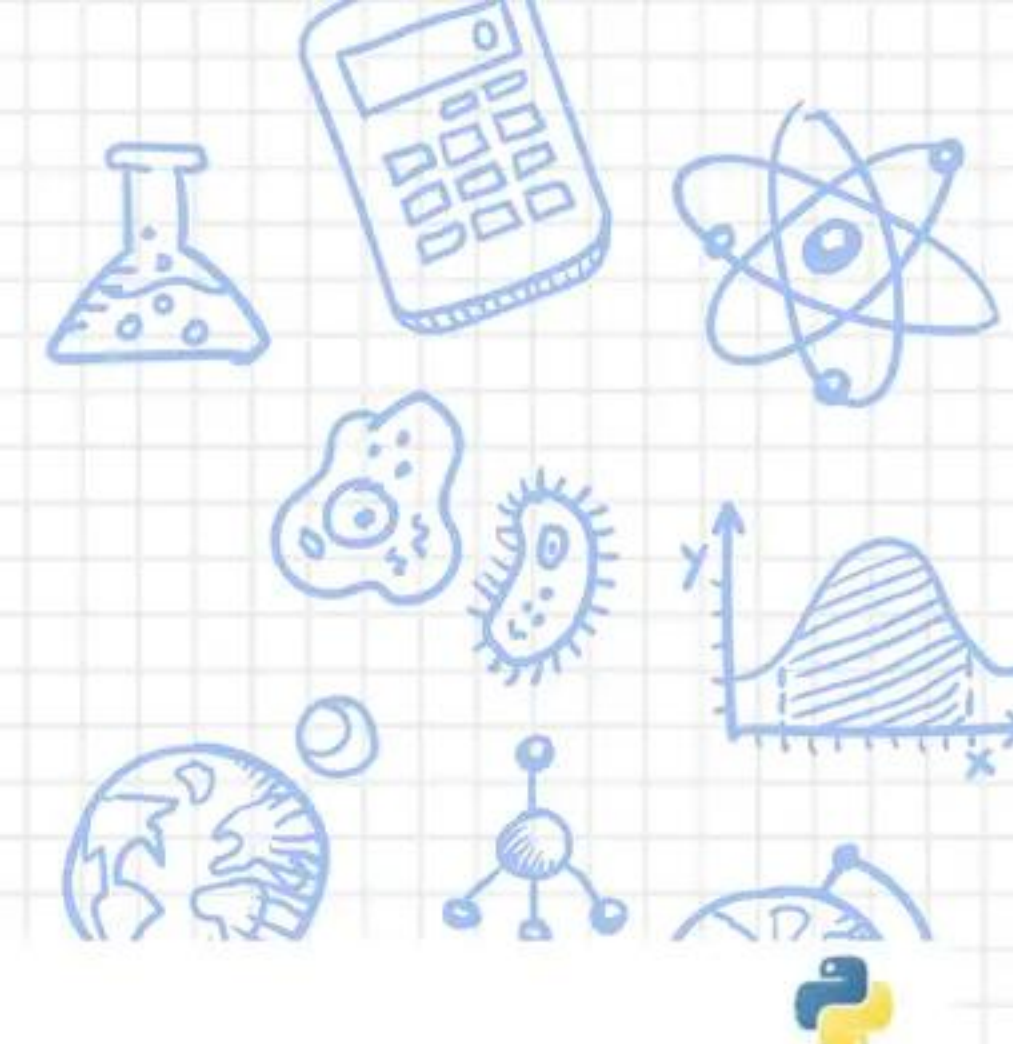
These are some common tasks you might perform when working with data:

- ✘ Acquire datasets that align with business needs
- ✘ Develop algorithms to transform data into useful, actionable information
- ✘ Build, test, and maintain database pipeline architectures
- ✘ Collaborate with management to understand company objectives
- ✘ Create new data validation methods and data analysis tools
- ✘ Ensure compliance with data governance and security policies









# Data Science Experiments

- ✘ Do the data understanding
- ✘ Data exploration
- ✘ Feature Engineering
- ✘ Method/Model Experiments

jupyter spectrogram (autosaved)

File Edit View Insert Cell Kernel Help | Python 3

+

Simple spectral analysis

An illustration of the [Discrete Fourier Transform](#)

$$X_k = \sum_{n=0}^{N-1} x_n \exp\left(\frac{-2\pi i}{N} kn\right) \quad k = 0, \dots, N-1$$

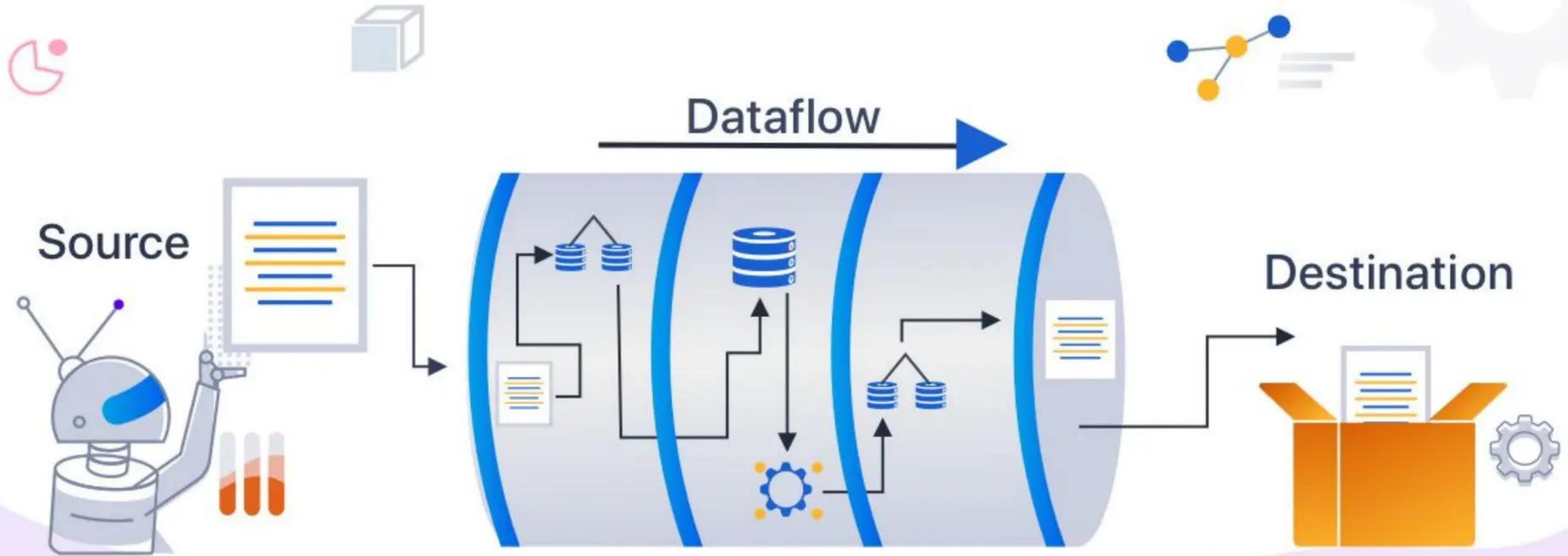
```
In [2]: from scipy.io import wavfile
rate, x = wavfile.read('test_mono.wav')
```

And we can easily view it's spectral structure using matplotlib's builtin specgram routine:

```
In [5]: fig, (ax1, ax2) = plt.subplots(1,2,figsize(16,5))
ax1.plot(x); ax1.set_title('Raw audio signal')
ax2.specgram(x); ax2.set_title('Spectrogram');
```

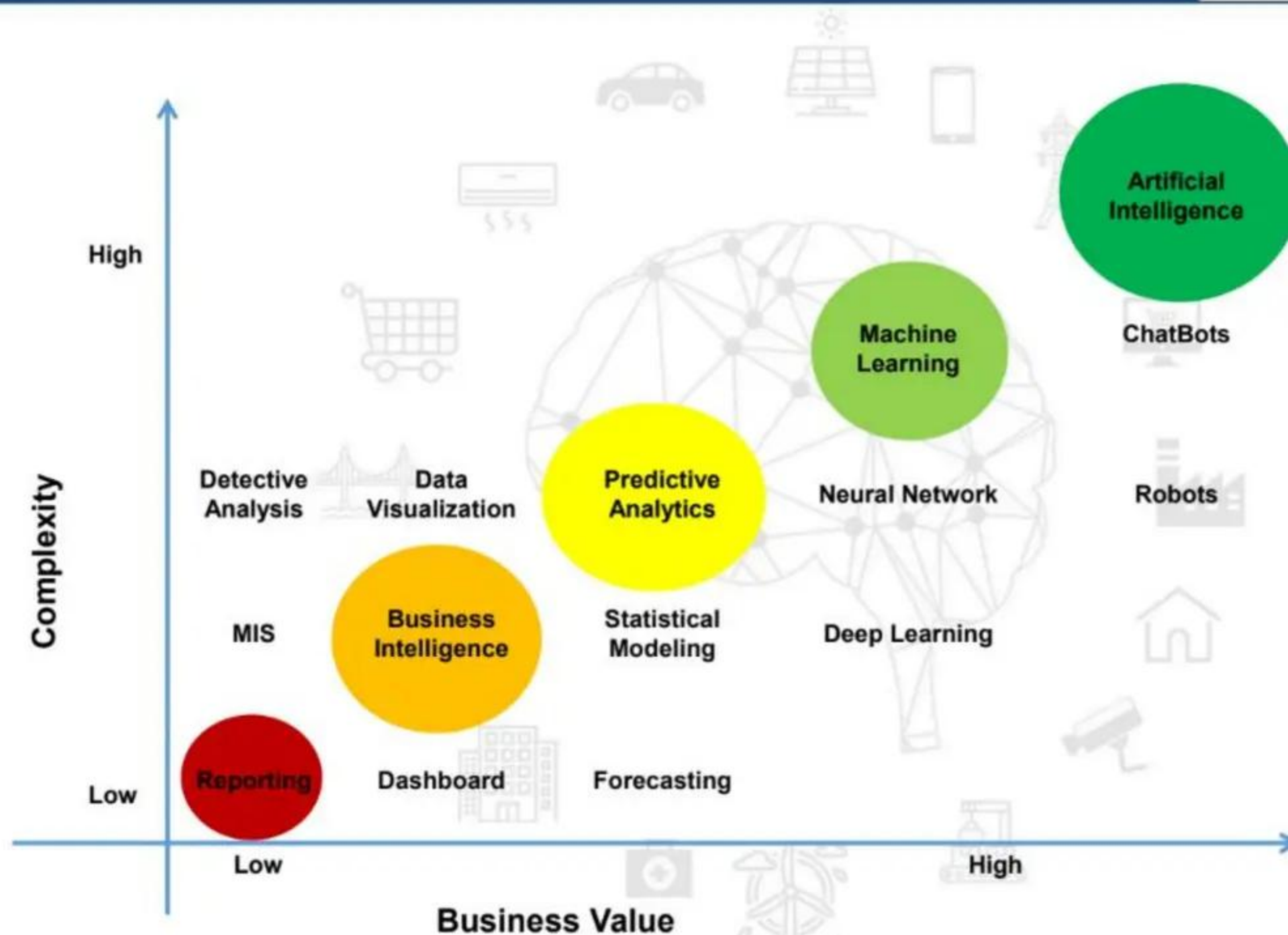


# Data flow

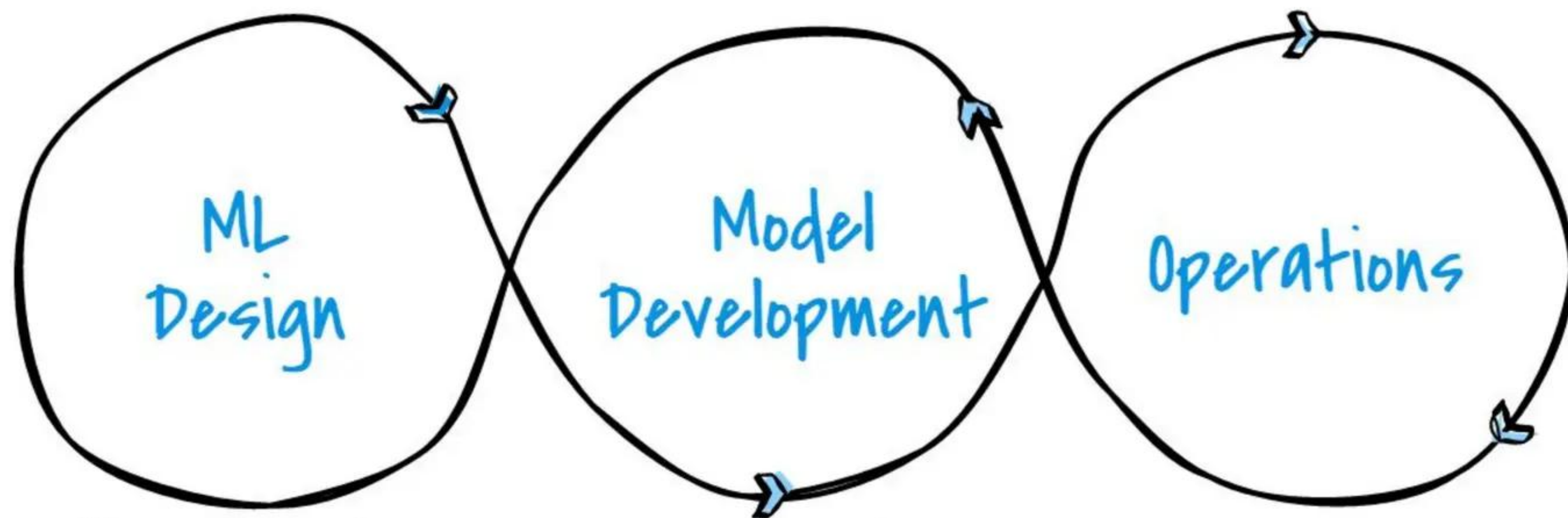




# Spectrum of Data Science



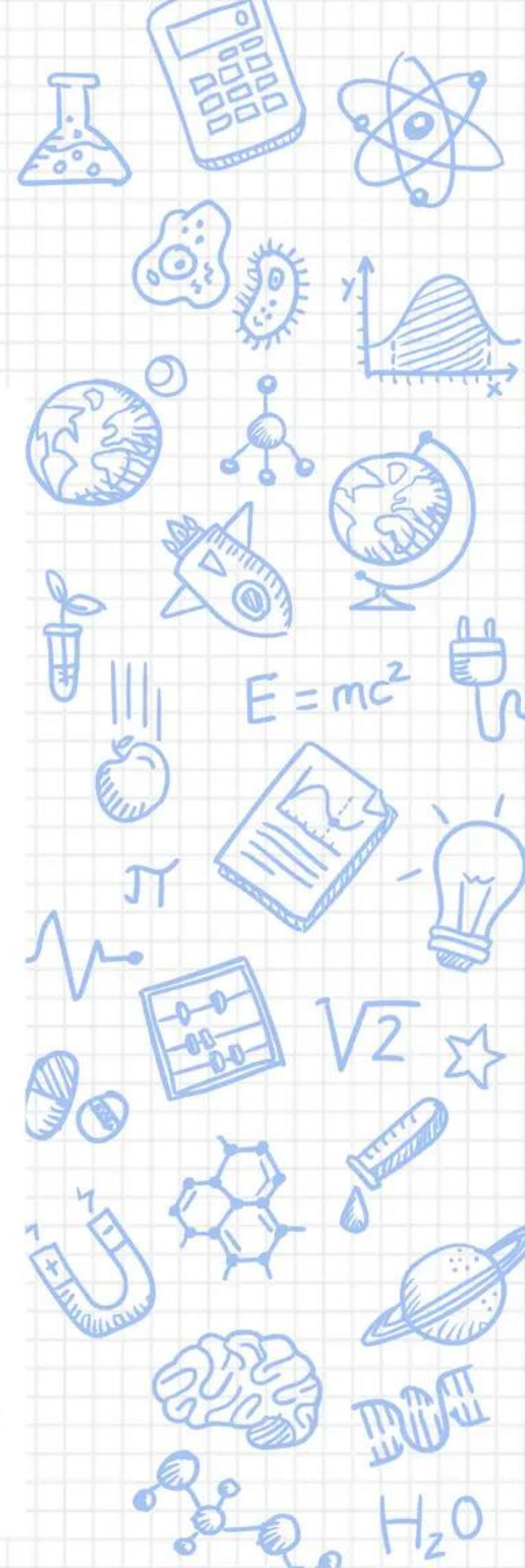
## Machine Learning Operations (MLOps)



- Gather requirements
- Prioritize ML use cases
- Business understanding
- Data Acquisition

- Data prep & processing
- Feature Engineering
- Model training / experimentation
- Model analysis & evaluation

- ML Model Deployment
- CI/CD Pipelines
- Model Monitoring & Triggering





Alation Amundsen collibra  
**METADATA MANAGEMENT**

acceldata MONTE CARLO SODA  
**OBSERVABILITY/QUALITY**

IMMUTA OKERA PRIVA  
**SECURITY/GOVERNANCE**

data.world Quilt atlan  
**DATA OPS**

Own{backup} Amazon Cloudwatch DATADOG  
**MAINTENANCE**

jupyter POPSQL MODE  
**COLLABORATION**

TACTON aporia ALGORITHMIA  
**ML OPS**

**DATA MANAGEMENT ACROSS THE PIPELINE**



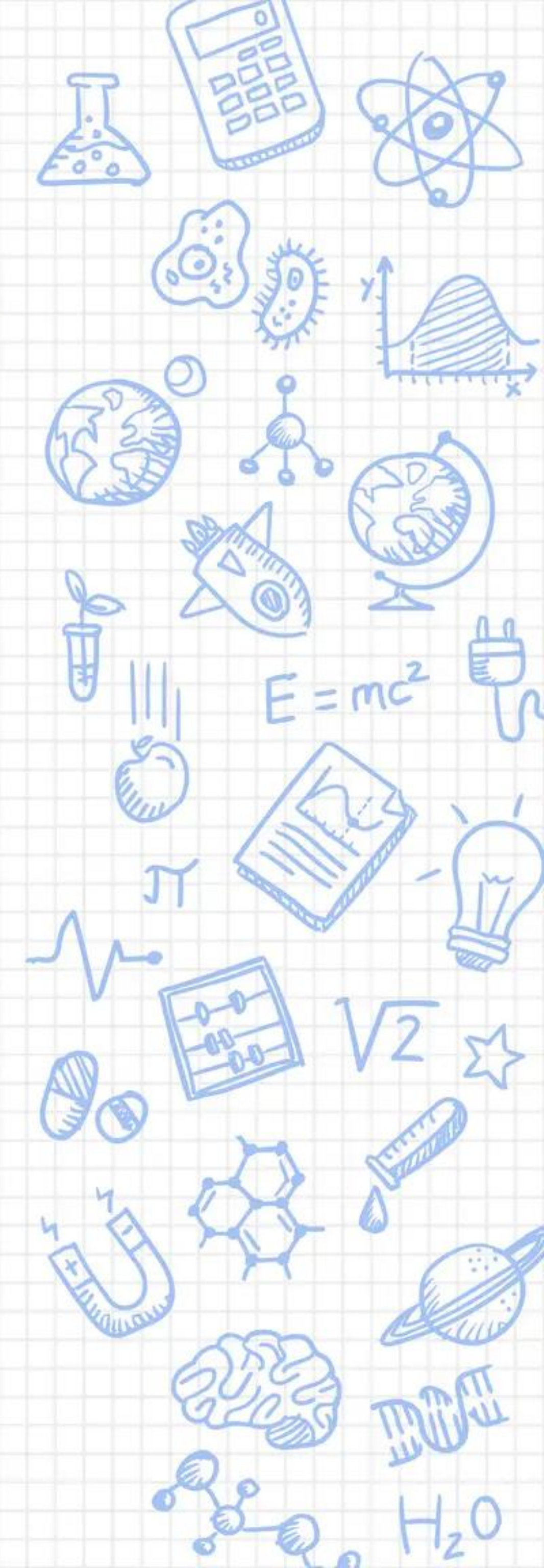
SAP Jira APPLICATIONS  
 Hadoop amazon S3 DBMS/FILE STORES  
 twilio stripe Google Maps 3RD PARTY APIS  
 SNOWFLOW Segment EVENT COLLECTORS  
 SUMO dynamtrace IBM Watson OTHERS

Xplenty talend FULL STACK ETL/ELT  
 Stitch Fivetran HEVO CONNECTORS  
 amazon KINESIS CONFLUENT EVENT STREAMING  
 dbt MATILLION lookML MODELLING  
 Spark Streaming Flink STREAM PROCESSING  
 DAGSTER Airflow WORKFLOW ORCHESTRATION  
 PREFECT

DELTA LAKE dremio upsolver DATA LAKES  
 amazon REDSHIFT snowflake Google BigQuery DATA WAREHOUSES  
 FILE/OBJECT STORES amazon S3 Microsoft Azure Blob Storage

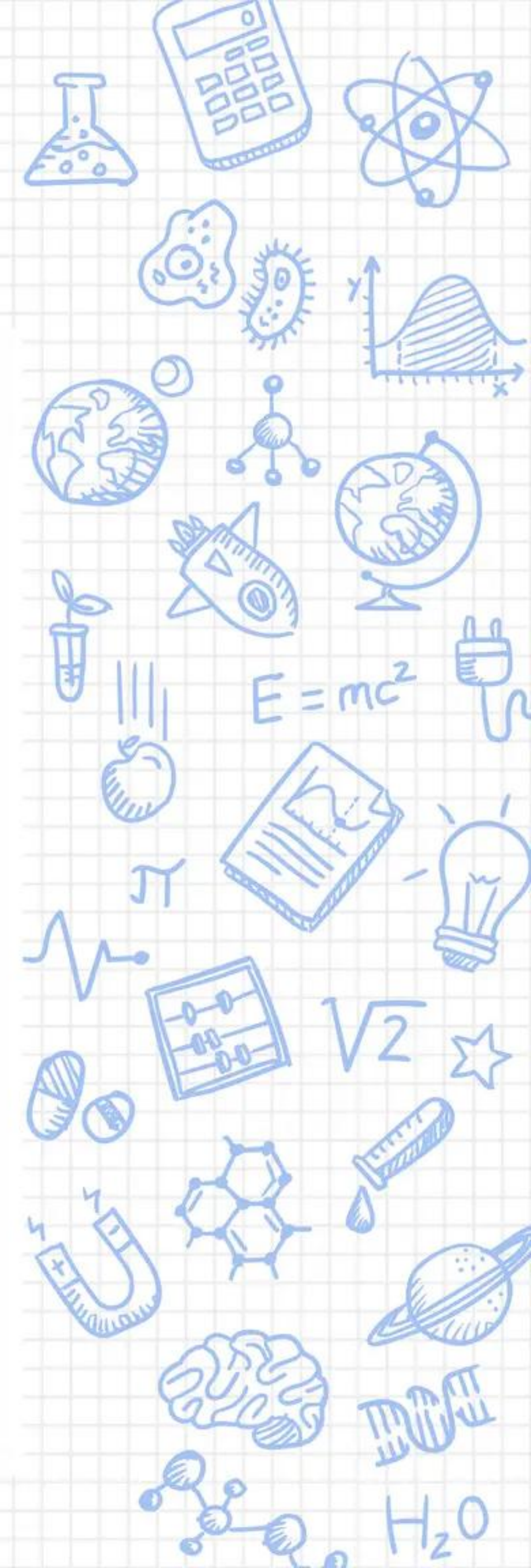
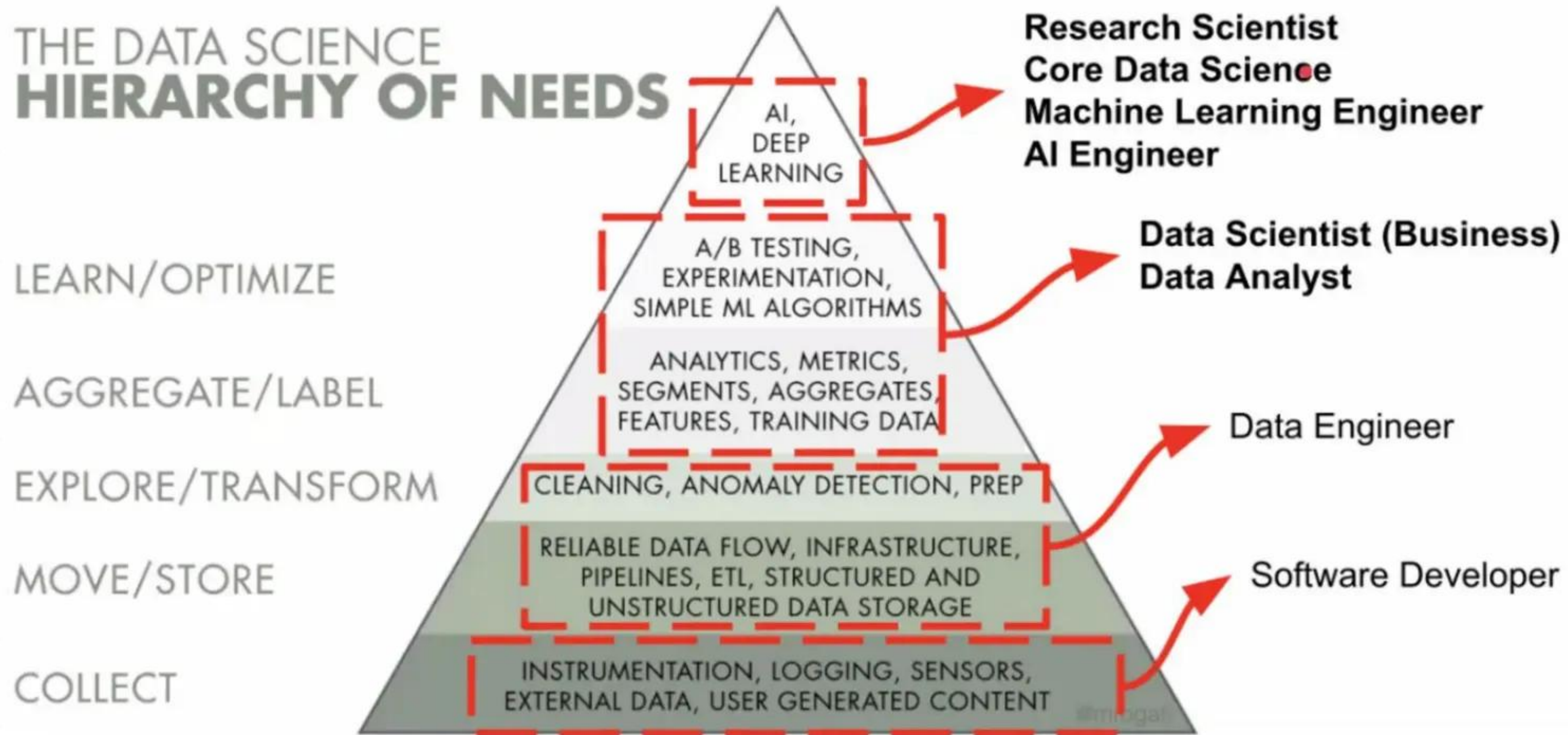
cloudera presto SQL-ON-HADOOP  
 ClickHouse druid LOW LATENCY  
 QUERYING PREPARATION  
 alteryx Power BI +tableau

tableau Power BI Looker BI TOOLS  
 anodot -outlier METRICS MONITORING  
 Streamlit Retool a\_ CUSTOM APPS  
 cube.js SISENSE EMBEDDED ANALYTICS  
 alteryx SAS python R DATA SCIENCE PLATFORMS  
 H2O.ai DataRobot AUTO ML PLATFORMS  
 BUSINESS INTELLIGENCE MACHINE LEARNING



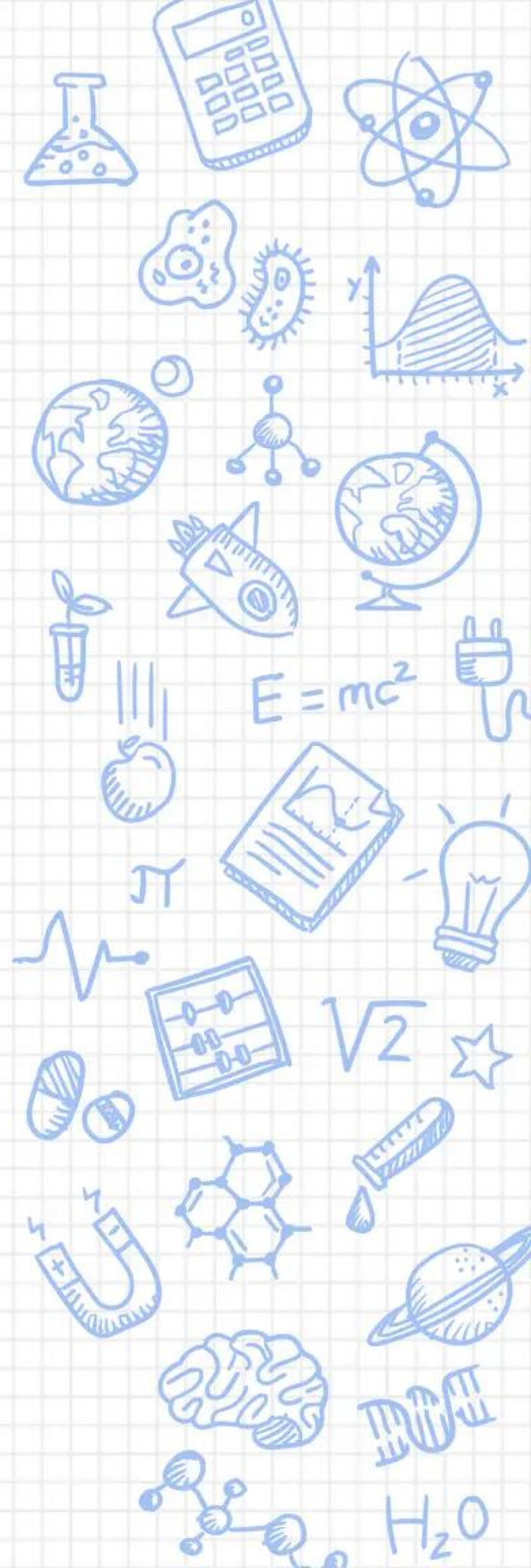
# Data Scientist work at Business Focused/Enterprise

## THE DATA SCIENCE HIERARCHY OF NEEDS







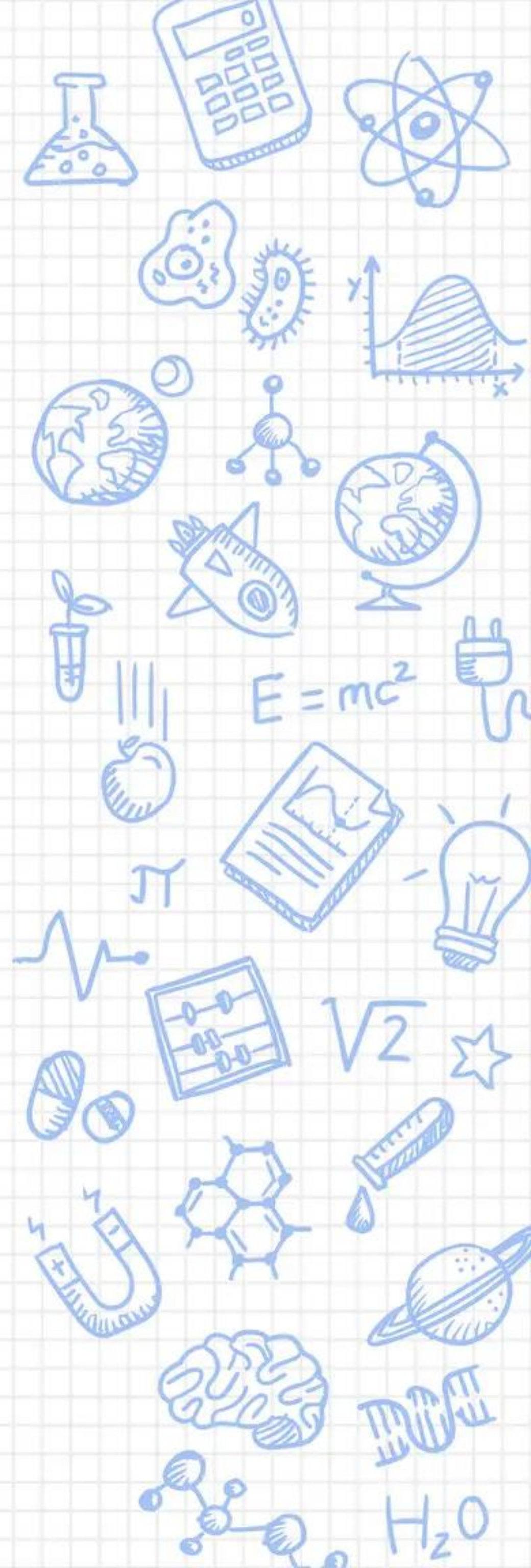
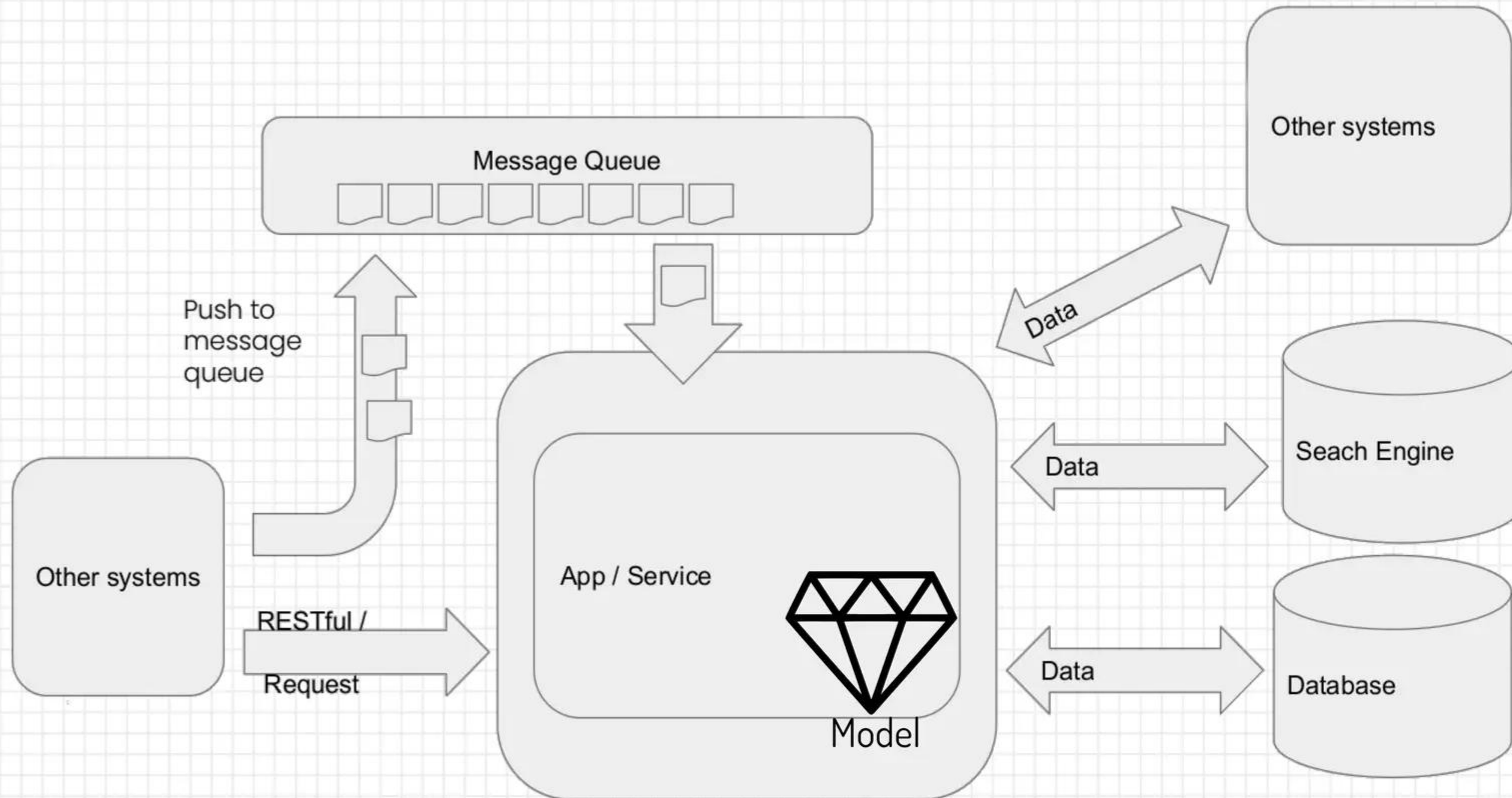


## Example of The Engine

---

- ✘ We need to run ML service
  - ✘ Sometimes for specific cases
  - ✘ The classification process is about 500ms
- ✘ We run the process (preprocess to main analytics process) as a batch process or need to process data with Large Volume
- ✘ The model is lightweight

# ML Service







## Model

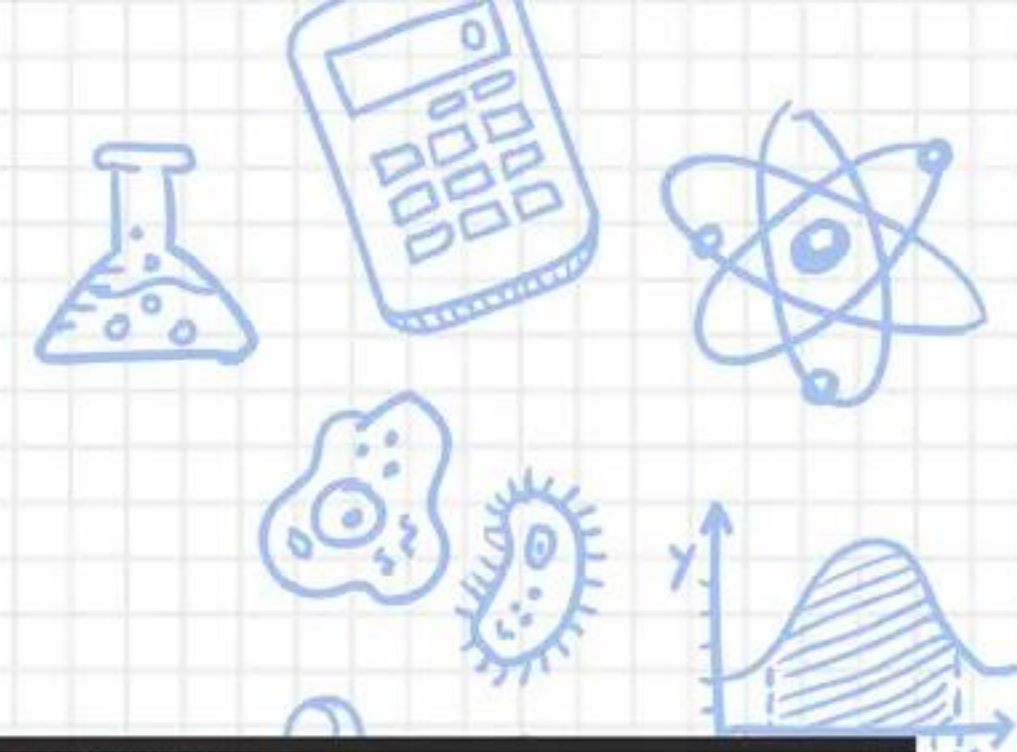
---

- ✗ We often use pickle or serializable file as model output
- ✗ The size bigger than it should be
- ✗ Impacted to the memory or the space complexity

```
total 44K
-rw-rw-r-- 1 hendri hendri 7,7K 0kt 22 08:24 model_irish_json.json
-rw-rw-r-- 1 hendri hendri 25K 0kt 22 08:41 model_irish_pk.model
-rw-rw-r-- 1 hendri hendri 778 0kt 22 08:41 sample_irish.py
drwxrwxr-x 5 hendri hendri 4,0K 0kt 22 08:27 venv
```



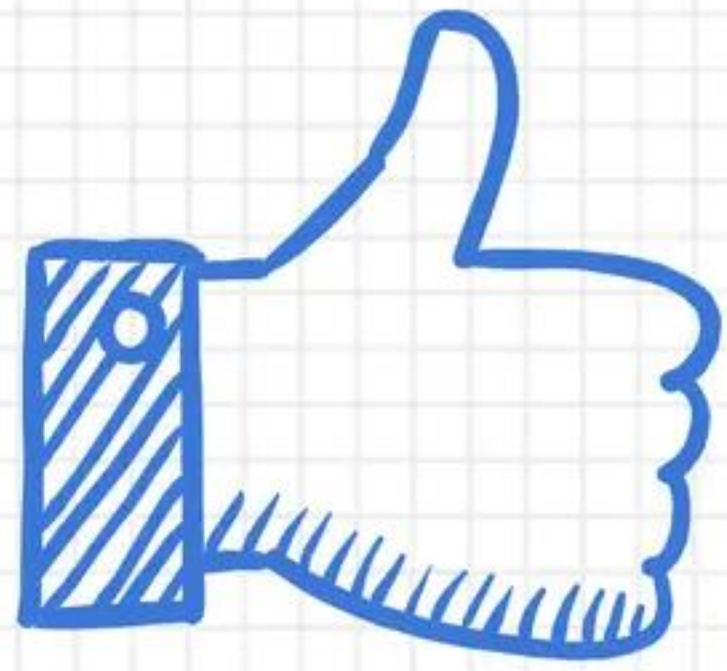




# Sample Model

```
1 {
2   "t": 30000,
3   "loss": 0.9975418627885244,
4   "coefs": [
5     [
6       -0.07013304935773942,
7       0.26527138138691336,
8       -0.4545038351870929,
9       -0.3110977212089695,
10      -0.4625050462952771,
11      -0.5343125797726832,
12      -0.4402206335231594,
13      -0.20220844801627716,
14      -0.13516308657082984,
15      0.14361541025115002
16    ],
17    [
18      -0.17309076443048899,
19      0.20877031353916853,
20      -0.38696278966005054,
21      0.5543423708181935,
22      -0.6107948147057291,
23      0.2192354029233313,
24      -0.07337205428792369,
25      0.07684301341968211,
26      -0.47084396112968563,
27      -0.33523524701191554
28    ],
29    [
30      0.47837749606938257,
31      0.5949163952784539,
32      -0.2442850620146144,
33      0.15116074637056276,
34      0.49280901728382515,
35      0.5163771269196054,
36      -0.6031703546008625,
37      0.6035180790935678,
38      -0.43229340027159696,
39      0.6071550915776449
40    ],
41    [
42      -0.4514688927311533,
43      -0.13142995975470284,
44      0.5995180466852053,
45      -0.05059343249410618,
46      0.2512260818048674,
47      -0.24165361644103628,
48      0.18983640904119806,
49      0.43812779279032404,
50      -0.6307086142307137,
51      0.44167737449606964
52    ],
53    [
54      -0.6646590688864646,
55      0.24756342987737248,
56      -0.38478655060000865
57    ],
58    [
59      -0.4859859296215193,
60      -0.02509408377991979,
61      -0.42578984771122774
62    ],
63    [
64      0.10070598214562193,
65      -0.4880012843362523,
66      0.12134231462532188
67    ],
68    [
69      0.28022007785909064,
70      -0.5807799688176638,
71      -0.08511792525114016
72    ],
73    [
74      0.26413776704302766,
75      -0.1166073956830679,
76      -0.61149275704756
77    ],
78    [
79      0.051372915328223745,
80      0.22122745040945643,
81      0.018956625883006974
82    ],
83    [
84      0.7093914871499977,
85      0.08908898113974999,
86      0.4713250978235561
87    ],
88    [
89      -0.4925748171332864,
90      -0.49012686651329174,
91      0.41766246260933054
92    ],
93    [
94      -0.13902978326423196,
95      -0.4546940504423227,
96      0.5808697026660242
97    ],
98    [
99      -0.2526457160902426,
100     0.3068346795771195,
101     0.38682285051572963
102    ],
103    [
104     "n_iter": 200,
105     "classes": {
106       "setosa",
107       "versicolor",
108       "virginica"
109     },
110     "n_layers": 3,
111     "best_loss": 0.9975418627885244,
112     "n_outputs": 3,
113     "intercepts": [
114       [
115         0.6406477882090572,
116         0.30359328159553006,
117         -0.2874662930420538,
118         0.3699381678270617,
119         -0.5194991024708189,
120         -0.07535424538033049,
121         0.5252238086483898,
122         -0.27022251069872527,
123         -0.27786730721840874,
124         -0.32852896685948774
125       ],
126       [
127         1.122728666292561,
128         1.122534200893611,
129         1.1222576150520247,
130         1.121907360628773,
131         1.1214917371694737,
132         1.1210185468215297,
133         1.1204946875232418,
134         1.1199264298773666,
135         1.1193194612747892,
136         1.1186789290532453,
137         1.1180094100419617,
138         1.117313062042723,
139         1.116594433684511,
140         1.1158565440447836,
141         1.115102902253569,
142         1.1143369414383,
143         1.1135604808260635,
144         1.1127740712560676,
145         1.1119801879484987,
146         1.1111804841278743,
147         1.1103779092749546,
148         1.1095714698103278,
149         1.1087645250703924,
150         1.1079577849766555,
151         1.1071525425096767,
152         1.1063493207580406,
153         1.1055491023819743,
154         1.1047508109065864,
155         1.1039556475642418,
156         1.1031648167307444,
157         1.1023771960512623,
158         1.1015959260993192,
159         1.1008203354765895,
160         1.1000505100916316,
161         1.0992869201216435,
162         1.0985297490460522,
163         1.0977791448403582,
164         1.0970340601651036,
165         1.096294608201062,
166         1.0955611626790853,
167         1.094834339722503,
168         1.0941121310010717,
169         1.0933961022642833,
170         1.0926865340613576,
171         1.0919834136749529,
172         1.091285527734828,
173         1.0905928989711482,
174         1.0899059706935836,
175         1.089221494596524,
176         1.0885430326217342,
177         1.0878712612151988,
178         1.0872031848077053,
179         1.0865401749616557,
180         1.0858823429931448,
181         1.0852273803987542,
182         1.0845754834198913,
183         1.08392864189277094,
184         1.08328649171502493,
185         1.08264917135278482,
186         1.0820167135278482,
187         1.081388628143069,
188         1.080763233365174,
189         1.0801417199373677,
190         1.0795245278857377,
191         1.078916024252967,
192         1.0783028835961168,
193         1.0776984167528574,
194         1.077098728291393,
195         1.076500896844384,
196         1.07590649019375,
197         1.0753158004081977,
198         1.074728560969761,
199         1.074144617819536,
200         1.0735635342488439,
201         1.0729840469486591,
202         1.0724077689686158,
203         1.0718346582294425,
204         1.0712646507955021,
205         1.070697691425527,
206         1.0701337226394352,
207         1.0695726851161764,
208         1.0690118056954172,
209         1.068450589556351,
210         1.0678911083832443,
211         1.0673338020220255,
212         1.0667784193407772,
213         1.0662252569653117,
214         1.0656742912376989,
215         1.0651254943285164,
216         1.0645786101500076,
217         1.0640335084400587,
218         1.0634894417339338,
219         1.062944759121887,
220         1.06239973696608,
221         1.0618536883932463,
222         1.0613087142799047,
223         1.0607629865273962,
224         1.0602183277329026,
225         1.0596749994734447,
226         1.0591329357137627,
227         1.0585914699374375,
228         1.0580507878393655,
229         1.0575093534332842,
230         1.0569686167086798,
231         1.0564288341728674,
232         1.055888132198703,
233         1.0553483285199396,
234         1.054806016625227,
235         1.0542598502259493,
236         1.053711821646284,
237         1.0531606935575872,
238         1.052605203729853,
239         1.0520489348520266,
240         1.0514902421702537,
241         1.0509291244313728,
242         1.0503677444918167,
243         1.04980473941957,
244         1.0492384161208417,
245         1.0486686653544866,
246         1.0480961411148981,
247         1.047523084488517,
248         1.033794006478145,
249         1.0331781440878856,
250         1.032561854064507,
251         1.0319414339979731,
252         1.0313177219846394,
253         1.0306902611953859,
254         1.0300613438842336,
255         1.0294316068733027,
256         1.0288011389708211,
257         1.0281760170662975,
258         1.0275383073780688,
259         1.02690060665755003,
260         1.0262733427882773,
261         1.025640176514014,
262         1.025006601434291,
263         1.0243726451482464,
264         1.0237383298319362,
265         1.0231036728308736,
266         1.0224686815011543,
267         1.0218383006752865,
268         1.0211914045456087,
269         1.0205520184041659,
270         1.0199121625870282,
271         1.0192718530158198,
272         1.01863101684313,
273         1.017989791064608,
274         1.017343523015222,
275         1.0166960952676183,
276         1.0160479191880467,
277         1.0153990288020884,
278         1.0147494528821197,
279         1.01409992147459393,
280         1.013448335949185,
281         1.012796824597725,
282         1.0121446994056567,
283         1.0114914687154761,
284         1.0108355671574019,
285         1.0101788978961277,
286         1.00952180547431,
287         1.008864055210001,
288         1.0082056067824783,
289         1.0075464668590417,
290         1.0068866396986165,
291         1.0062261274298984,
292         1.0055649302936622,
293         1.0049028546692154,
294         1.0042379075256607,
295         1.0035721245025346,
296         1.0029054937438926,
297         1.002238024172403,
298         1.0015697223160416,
299         1.00090059257184,
300         1.0002306374493781,
301         0.9995597653638649,
302         0.9988879677292097,
303         0.9982153341094416,
304         0.9975418627885244
305     ]
306   ],
307   "n_features_in": 4,
308   "out_activation": "softmax",
309   "label_binarizer": "LabelBinarizer",
310   "validation_scores": null,
311   "best_validation_score": null
312 }
313 ]
314 }
```





# THANKS!

## Any questions?

You can find me at

- ✘ X @infoHendri
- ✘ Telegram @siganteng
- ✘ Email:

[situkangsayur@gmail.com](mailto:situkangsayur@gmail.com)

[hendri.karisma@akarintidata.ai](mailto:hendri.karisma@akarintidata.ai)