```
!unzip artinsight_poc_mockai.zip -d artinsight_poc
```

```
Archive:  artinsight_poc_mockai.zip
  inflating: artinsight_poc/README.md
  inflating: artinsight_poc/LICENSE
  inflating: artinsight_poc/src/monitoring.py
  inflating: artinsight_poc/src/analysis.py
  inflating: artinsight_poc/src/scoring.py
  inflating: artinsight_poc/src/analytics.py
  inflating: artinsight_poc/src/utils.py
  inflating: artinsight_poc/src/dashboard.py
  inflating: artinsight_poc/docs/architecture.md
  inflating: artinsight_poc/docs/algorithms.md
  inflating: artinsight_poc/notebooks/poc_colab_instructions.md
  inflating: artinsight_poc/notebooks/poc_starter.py
```

```
!pip install streamlit matplotlib pandas pyngrok
```

```
Requirement already satisfied: streamlit in
/usr/local/lib/python3.12/dist-packages (1.52.0)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: pandas in
/usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: pyngrok in
/usr/local/lib/python3.12/dist-packages (7.5.0)
Requirement already satisfied: altair!=5.4.0,!=5.4.1,<7,>=4.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (5.5.0)
Requirement already satisfied: blinker<2,>=1.5.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (1.9.0)
Requirement already satisfied: cachetools<7,>=4.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (6.2.2)
Requirement already satisfied: click<9,>=7.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (8.3.1)
Requirement already satisfied: numpy<3,>=1.23 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (2.0.2)
Requirement already satisfied: packaging>=20 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (25.0)
Requirement already satisfied: pillow<13,>=7.1.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (11.3.0)
Requirement already satisfied: protobuf<7,>=3.20 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (5.29.5)
Requirement already satisfied: pyarrow>=7.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (18.1.0)
Requirement already satisfied: requests<3,>=2.27 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (2.32.4)
Requirement already satisfied: tenacity<10,>=8.1.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (9.1.2)
Requirement already satisfied: toml<2,>=0.10.1 in
```

```
/usr/local/lib/python3.12/dist-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.4.0 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (4.15.0)
Requirement already satisfied: watchdog<7,>=2.1.5 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (6.0.0)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (3.1.45)
Requirement already satisfied: pydeck<1,>=0.8.0b4 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (0.9.1)
Requirement already satisfied: tornado!=6.5.0,<7,>=6.0.3 in
/usr/local/lib/python3.12/dist-packages (from streamlit) (6.5.1)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.12/dist-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.12/dist-packages (from matplotlib)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: PyYAML>=5.1 in
/usr/local/lib/python3.12/dist-packages (from pyngrok) (6.0.3)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.12/dist-packages (from altair!=5.4.0,!
=5.4.1,<7,>=4.0->streamlit) (3.1.6)
Requirement already satisfied: jsonschema>=3.0 in
/usr/local/lib/python3.12/dist-packages (from altair!=5.4.0,!
=5.4.1,<7,>=4.0->streamlit) (4.25.1)
Requirement already satisfied: narwhals>=1.14.2 in
/usr/local/lib/python3.12/dist-packages (from altair!=5.4.0,!
=5.4.1,<7,>=4.0->streamlit) (2.12.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in
/usr/local/lib/python3.12/dist-packages (from gitpython!
=3.1.19,<4,>=3.0.7->streamlit) (4.0.12)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.17.0)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests<3,>=2.27-
>streamlit) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in
```

```
/usr/local/lib/python3.12/dist-packages (from requests<3,>=2.27-
>streamlit) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.12/dist-packages (from requests<3,>=2.27-
>streamlit) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.12/dist-packages (from requests<3,>=2.27-
>streamlit) (2025.11.12)
Requirement already satisfied: smmap<6,>=3.0.1 in
/usr/local/lib/python3.12/dist-packages (from gitdb<5,>=4.0.1-
>gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.0.2)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.12/dist-packages (from jinja2->altair!=5.4.0,!
=5.4.1,<7,>=4.0->streamlit) (3.0.3)
Requirement already satisfied: attrs>=22.2.0 in
/usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!
=5.4.0,!=5.4.1,<7,>=4.0->streamlit) (25.4.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!
=5.4.0,!=5.4.1,<7,>=4.0->streamlit) (2025.9.1)
Requirement already satisfied: referencing>=0.28.4 in
/usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!
=5.4.0,!=5.4.1,<7,>=4.0->streamlit) (0.37.0)
Requirement already satisfied: rpds-py>=0.7.1 in
/usr/local/lib/python3.12/dist-packages (from jsonschema>=3.0->altair!
=5.4.0,!=5.4.1,<7,>=4.0->streamlit) (0.29.0)
```

```
import os
import json
import pandas as pd
from dash import Dash, dcc, html, Input, Output
from pyngrok import ngrok
import nest_asyncio
import threading
import subprocess
import signal
import time

# Apply asyncio patch for Colab
nest_asyncio.apply()

# □ Set working directory (adjust if needed)
os.chdir('/content/artinsight_poc')

# ı Ensure sample data exists
sample_file = "sample_opportunities.json"

if not os.path.exists(sample_file):
    # If you have your utils/scoring modules, generate sample data
    from src import monitoring, analysis, scoring, utils
    posts = monitoring.fetch_reddit(mock=True, count=5)
    posts += monitoring.fetch_pinterest(mock=True, count=5)

    scored = []
    for p in posts:
        p['score'] = scoring.score_opportunity(p)
        p['suggestion'] = analysis.generate_suggestion(p)
        p['approved'] = False
        scored.append(p)

    utils.save_sample(scored, path=sample_file)
    print(f"□ Generated {sample_file} with {len(scored)} items")
else:
    print(f"□ {sample_file} already exists")

# 2 Load sample data
with open(sample_file) as f:
    data = json.load(f)
df = pd.DataFrame(data)

# 3 Kill old ngrok tunnels
ngrok.kill()
print("□ Old ngrok tunnels stopped")

# 4 Kill any processes using port 8050
try:
    result = subprocess.check_output("lsof -t -i:8050",
shell=True).decode().split()
    for pid in result:
        try:
            os.kill(int(pid), signal.SIGTERM)
        except ProcessLookupError:
            pass
    if result:
        print("□ Killed old processes on port 8050")
except subprocess.CalledProcessError:
    print("□ No processes found on port 8050")

# 5 Create Dash app
app = Dash(__name__)
app.layout = html.Div([
    html.H1("ArtInsight POC Dashboard"),
    html.Label("Select Post:"),
    dcc.Dropdown(
        id="select-post",
        options=[{"label": p["text"], "value": i} for i, p in
df.iterrows()],
        value=0
    ),
    html.Br(),
    html.Div(id="post-details")
])

@app.callback(
    Output("post-details", "children"),
    Input("select-post", "value")
)
def show_post_details(idx):
    post = df.iloc[idx]
    return html.Div([
        html.P(f"Title: {post['text']}"),
        html.P(f"Score: {post['score']}"),
        html.P(f"Suggestion: {post['suggestion']}"),
        html.P(f"Approved: {post['approved']}")
    ])

#6 6 Run Dash app in a separate thread
def run_dash_app():
    app.run(port=8050, debug=False)

threading.Thread(target=run_dash_app, daemon=True).start()

#7 7 Wait for server to start
time.sleep(5)

#8 8 Set ngrok auth token
ngrok.set_auth_token("36OLnCkKP5EJca0oqrVCDpVnOQ9_nrqWqnX6Vh2rzjJrnE6n
")  # replace with your token

#9 9 Open ngrok tunnel with pooling
public_url = ngrok.connect(8050, options={"pooling_enabled": True})
print(f"□ Dash app is live at: {public_url}")
```

```
□ sample_opportunities.json already exists
□ Old ngrok tunnels stopped
□ No processes found on port 8050

<IPython.core.display.Javascript object>

WARNING:pyngrok.process.ngrok:t=2025-12-04T19:27:01+0000 lvl=warn
msg="invalid tunnel configuration" pg=/api/tunnels id=14bff02b9d6ee89d
err="yaml: unmarshal errors:\n  line 1: field options not found in
type config.HTTPv2Tunnel"

----------------------------------------------------------------------
-----
HTTPError                                 Traceback (most recent call
last)
/usr/local/lib/python3.12/dist-packages/pyngrok/ngrok.py in
api_request(url, method, data, params, timeout, auth)
```

```
    573     try:
--> 574         response = urlopen(request, encoded_data, timeout)
    575         response_data = response.read().decode("utf-8")

/usr/lib/python3.12/urllib/request.py in urlopen(url, data, timeout,
cafile, capath, cadefault, context)
    214         opener = _opener
--> 215     return opener.open(url, data, timeout)
    216

/usr/lib/python3.12/urllib/request.py in open(self, fullurl, data,
timeout)
    520             meth = getattr(processor, meth_name)
--> 521             response = meth(req, response)
    522

/usr/lib/python3.12/urllib/request.py in http_response(self, request,
response)
    629         if not (200 <= code < 300):
--> 630             response = self.parent.error(
    631                 'http', request, response, code, msg, hdrs)

/usr/lib/python3.12/urllib/request.py in error(self, proto, *args)
    558             args = (dict, 'default', 'http_error_default') +
orig_args
--> 559             return self._call_chain(*args)
    560

/usr/lib/python3.12/urllib/request.py in _call_chain(self, chain,
kind, meth_name, *args)
    491             func = getattr(handler, meth_name)
--> 492             result = func(*args)
    493             if result is not None:

/usr/lib/python3.12/urllib/request.py in http_error_default(self, req,
fp, code, msg, hdrs)
    638     def http_error_default(self, req, fp, code, msg, hdrs):
--> 639         raise HTTPError(req.full_url, code, msg, hdrs, fp)
    640

HTTPError: HTTP Error 400: Bad Request

During handling of the above exception, another exception occurred:

PyngrokNgrokHTTPError                     Traceback (most recent call
last)
/tmp/ipython-input-3308683493.py in <cell line: 0>()
    100
9   101 # 9 Open ngrok tunnel with pooling
--> 102 public_url = ngrok.connect(8050, options={"pooling_enabled":
```

```
True})
    103 print(f"□ Dash app is live at: {public_url}")

/usr/local/lib/python3.12/dist-packages/pyngrok/ngrok.py in
connect(addr, proto, name, pyngrok_config, **options)
    337     logger.debug(f"Creating tunnel with options: {options}")
    338
--> 339     tunnel = NgrokTunnel(api_request(f"{api_url}/api/tunnels",
method="POST", data=options,
    340
timeout=pyngrok_config.request_timeout),
    341                           pyngrok_config, api_url)

/usr/local/lib/python3.12/dist-packages/pyngrok/ngrok.py in
api_request(url, method, data, params, timeout, auth)
    593         logger.debug(f"Response {status_code}:
{response_data.strip()}")
    594
--> 595         raise PyngrokNgrokHTTPError(f"ngrok client exception,
API returned {status_code}: {response_data}",
    596                                     e.url,
    597                                     status_code, e.reason,
e.headers, response_data)

PyngrokNgrokHTTPError: ngrok client exception, API returned 400:
{"error_code":102,"status_code":400,"msg":"invalid tunnel
configuration","details":{"err":"yaml: unmarshal errors:\n  line 1:
field options not found in type config.HTTPv2Tunnel"}}
```