

Case Study 概述

一：Yup 项目研究背景和目标

1.1 项目背景

本项目源于一次真实的广告投放实践。在代理团队为金融科技公司 **Finture Tech** 旗下的 **YUP Visa** 信用卡于印尼市场进行 Google、TikTok、Meta 三渠道的 App 广告冷启动时，我观察到了两个与行业常规经验相悖的“异常”现象：

1. 成本悖论：当参考印尼地区其他信贷/金融产品设置行业平均水平的 CPA（单次获客成本）目标时，广告花费极高且不稳定；而将 CPA 目标大幅下调至行业均值的一半以下时，消耗反而变得平稳。这暗示 YUP 的转化用户对成本极其敏感，或与常规信贷用户画像不同。
2. 素材表现悖论：
 - **TikTok/Meta** 渠道：客户提供的“历史跑量素材”视频风格极简陋（如人手划过写有贷款信息的纸张），这与金融产品通常追求信任、专业的素材风格大相径庭。这些素材互动率高但转化差，吸引了大量“看热闹”的非目标人群。
 - **Google** 渠道：表现最佳的素材并非金融场景，而是“炸鸡/甜甜圈”等食物图片+直接文案的组合。这类素材在初期以极低的成本（7-12 美元）稳定获取了符合要求的 App 授信转化。
3. **Yup** 特殊广告语：在谷歌搜索 Finture Tech 公司官网或者 Yup 信用卡产品可以看到广告标题文案是：

For those who've been told NO we say YUP

1.2 核心假设与猜测

基于上述现象，我形成了一个核心猜测：**YUP** 信用卡的目标受众极为特殊——他们不是主动寻求贷款的“传统借款人”，而是那些“被传统银行拒绝、对正规金融产品缺乏了解、但又有稳定收入的工薪阶层（Middle/Lower-Middle Class）”。

这导致：

- 在主动探索型平台（如 TikTok/Meta），通投“贷款人群”会错误覆盖大量非目标客群，而简陋的“猎奇”素材进一步吸引了错误流量。
- 在意图搜索型平台（如 Google），简单的食物素材可能恰好切中了用户“改善生活”的即时、朴实需求，并在线下合作（如快餐店）与熟人推荐的配合下，意外高效地触达了真实目标用户。

1.3 项目目标

为了验证以上猜测，并指导未来的市场策略，我决定跳出广告后台的局限，发起本次独立的案头研究项目，旨在：

1. 产品与市场分析：从宏观层面理解，为何 YUP 这款产品能在 2025 年的印尼市场获得成功？它究竟切中了怎样的市场空白？
2. 渠道表现复盘：系统性分析，为何在冷启动期，Meta/TikTok 渠道的常规打法失效，而 Google 的“非典型”素材却成功了？背后反映了怎样的用户行为差异？
3. 策略指导：基于数据洞察，为 YUP 未来在印尼市场的品牌宣传与预算分配，提供一个初步的数据支持框架。

二：印尼地区宏观市场环境

宏观环境 PEST 分析 + 市场机会量化评估

P - Political : 信贷法律制度 (Section 2.2)

E - Economic : 收入水平与分配 (Section 2.4)

S - Social : 金融包容性现状 (Section 2.1)

T - Technological : 数字基础设施 (Section 2.3)

2.1 金融包容性现状

- 东南亚地区（中等教育以上）成年人银行账户持有率
- 印尼（中等教育以上）成年人银行账户持有率近五年趋势
- 总结：印尼 Yup 市场规模- 万潜在用户约 2-3 千万人

2.1.1 画图 (Social)

```
# 1. 运行这个单元格来上传文件
from google.colab import files
import pandas as pd

# 点击运行后，会弹出文件选择窗口
## 上传文件'account 持有率东南亚Data-世界银行.csv'
uploaded = files.upload()

# 查看上传的文件名
print("已上传的文件:", list(uploaded.keys()))

<IPython.core.display.HTML object>

Saving account 持有率东南亚Data-世界银行.csv to account 持有率东南亚Data-世界
银行.csv
已上传的文件: ['account 持有率东南亚Data-世界银行.csv']
```

```
# 2. 读取CSV文件
# 假设你的文件名为world_bank_data.csv
file_name = list(uploaded.keys())[0] # 获取第一个文件名

# 读取数据，跳过可能的空行
df = pd.read_csv(file_name, skip_blank_lines=True)

# print("原始数据形状:", df.shape)
# print("\n前3行数据:")
# print(df.head(3))
# print("\n列名:")
# print(df.columns.tolist())

# 3. 数据清洗 - 针对世界银行数据的常见问题
def clean_worldbank_data(df):
    """
    专门清洗世界银行格式的数据
    """

    # 删除全为空值的行
    df_clean = df.dropna(how='all')

    # 删除描述行（通常包含'Data from database'等文本）
    mask = df_clean.iloc[:, 0].astype(str).str.contains('Data from database|Last Updated', na=False)
    df_clean = df_clean[~mask].reset_index(drop=True)

    # 重命名列，去除年份列的'[YRXXXX]'后缀
    df_clean.columns = [col.split(' ')[0] if ' ' in str(col) else col for col in df_clean.columns]

    # 替换'...'为NaN
    df_clean = df_clean.replace('...', pd.NA)

    # 转换数值列
    year_columns = [str(year) for year in range(2015, 2025)]
    for col in year_columns:
        if col in df_clean.columns:
            df_clean[col] = pd.to_numeric(df_clean[col], errors='coerce')

    return df_clean

# 应用清洗函数
df_clean = clean_worldbank_data(df)

# print("清洗后数据形状:", df_clean.shape)
# print("\n清洗后数据预览:")
# print(df_clean.head())
# print("\n数据类型:")
# print(df_clean.dtypes)
```

```

# 4. 提取我们关心的数据(东南亚五国)
target_countries = ['Indonesia', 'Philippines', 'Singapore',
                    'Thailand', 'Malaysia']

# 筛选国家
df_selected = df_clean[df_clean['Country
Name'].isin(target_countries)].copy()

print("筛选出的国家数据:")
print(df_selected[['Country Name', '2017', '2021', '2024']])

筛选出的国家数据:
   Country Name    2017    2021    2024
0  Indonesia  62.723114  61.894173  68.327671
1  Philippines  41.352767  61.175079  57.903109
2  Singapore  99.058811  98.961243  98.680383
3  Thailand   86.625561  97.311675  93.933271
4  Malaysia   87.535507  92.941635  92.453429

import plotly.graph_objects as go
from plotly.subplots import make_subplots
df = df_selected

# 创建Plotly图表 - 调整尺寸
fig = make_subplots(
    rows=1, cols=2,
    subplot_titles=( '2024年各国金融账户持有率对比',
                     '印尼金融包容性变化趋势(2017-2024)' ),
    specs=[[{"type": "bar"}, {"type": "scatter"}]],
    horizontal_spacing=0.15, # 增加子图之间的水平间距
    vertical_spacing=0.1      # 增加子图之间的垂直间距
)

# 图表1：柱状图
countries = df['Country Name'].tolist()
values_2024 = df['2024'].tolist()

colors = ['red' if c == 'Indonesia' else 'lightblue' for c in
countries]

fig.add_trace(
    go.Bar(
        x=countries,
        y=values_2024,
        marker_color=colors,
        text=[f'{v:.1f}%' for v in values_2024],
        textposition='auto',
        name='2024 Account Ownership',
        textfont=dict(size=10) # 调整文字大小
),

```

```

        row=1, col=1
    )

# 图表2：折线图
years = ['2017', '2021', '2024']
indo_values = df[df['Country Name'] == 'Indonesia'][years].values[0]

fig.add_trace(
    go.Scatter(
        x=years,
        y=indo_values,
        mode='lines+markers+text',
        line=dict(color='red', width=2.5), # 稍微减小线宽
        marker=dict(size=8, color='white', line=dict(color='red',
width=1.5)), # 减小标记大小
        text=[f'{v:.1f}%' for v in indo_values],
        textposition='top center',
        textfont=dict(size=10), # 调整文字大小
        name='Indonesia Trend'
    ),
    row=1, col=2
)

# 更新布局 - 调整为更适合 Colab 和 PDF 的尺寸
fig.update_layout(
    title_text='世界银行数据：印尼金融包容性分析',
    showlegend=False,
    height=400, # 减小高度
    width=850, # 减小宽度，适合 Colab 显示
    margin=dict(l=50, r=50, t=80, b=50), # 调整边距
    title_font=dict(size=18), # 调整标题字体大小
    font=dict(size=11) # 调整整体字体大小
)

# 更新子图标题样式
fig.update_annotations(font_size=14)

# 更新坐标轴
fig.update_xaxes(title_text="国家", row=1, col=1,
tickfont=dict(size=10))
fig.update_yaxes(title_text="账户持有率 (%)", row=1, col=1, range=[0,
105])
fig.update_xaxes(title_text="年份", row=1, col=2)
fig.update_yaxes(title_text="账户持有率 (%)", row=1, col=2, range=[50,
75])

# 计算增长
growth_2021_2024 = indo_values[2] - indo_values[1]
growth_text = f"+{growth_2021_2024:.1f}%" if growth_2021_2024 >= 0

```

```

else f"{{growth_2021_2024:.1f}}%"

# 调整注释位置和大小
fig.add_annotation(
    x=1.5, # 2021 和 2024 中间
    y=(indo_values[1] + indo_values[2]) / 2 + 0.8, # 调整位置
    text=f"增长: {{growth_text}}",
    showarrow=True,
    arrowhead=2,
    arrowsize=1,
    arrowwidth=1.5,
    arrowcolor="green" if growth_2021_2024 > 0 else "red",
    row=1,
    col=2,
    font=dict(size=10) # 调整注释字体大小
)

# 显示图表
fig.show()

# # 同时, 这里提供PDF 导出建议
# print("提示: 如需导出PDF, 建议使用以下方法:")
# print("1. 将图表保存为图片: fig.write_image('financial_inclusion.png', scale=2)")
# print("2. 在Colab中可以使用:")
# print("from google.colab import files")
# print("files.download('chart.png')")
# print("3. 然后将图片插入到PDF 文档中")

# 5. 打印关键数据
print("关键发现:")
print("*"*50)
indo_2024 = df[df['Country Name'] == 'Indonesia']['2024'].values[0]
print(f"1. 2024年印尼中等教育以上人群中, 仍有 {{100-indo_2024:.1f}}% 没有正规金融账户")
print(f"2. 这正是Yup信用卡的目标市场: {{100-indo_2024:.1f}}%的潜在用户")
print(f"3. 相比2021年, 2024年印尼覆盖率增长: {{indo_values[2]-indo_values[1]:+.1f}}%")
print("*"*50)

```

关键发现：

- ```
=====
1. 2024年印尼中等教育以上人群中, 仍有 31.7% 没有正规金融账户
2. 这正是Yup信用卡的目标市场: 31.7%的潜在用户
3. 相比2021年, 2024年印尼覆盖率增长: +6.4%
=====
```

## 2.1.2 社会现状洞察：印尼 Yup 市场规模- 万潜在用户约 2-3 千万人

### 计算过程

1. 印尼总人口：印尼大约 2.8 亿人（世界银行 2024 年数据）
2. 其中的成年人：其中约 68% 是 15 岁以上成年人  $\approx 1.9$  亿成年人
3. 教育水平：这些成年人中约 65% 有中等以上教育  $\approx 1.25$  亿人
4. 金融状况：数据显示，这些人中 31.7% 没有正规银行账户
5. 计算： $1.25 \text{ 亿} \times 31.7\% \approx 3960 \text{ 万人}$

### 估算 2-3 千万人的区间

- 这是保守估计，扣除了部分不适合信贷的人群，比如 15-20 岁人群/还贷能力有风险人群
- 考虑到数据统计的误差
- 便于记忆和传播

### 数据来源

- 总人口：世界银行
- 成年人比例：联合国人口司
- 教育水平：印尼统计局(BPS)
- 金融账户数据：世界银行

总结：在印尼，大约每 10 个成年人中，就有至少 1 个“受过中等教育、有收入但没银行账户”，这些人正是 Yup 的目标用户。

## 2.2 信贷法律制度

- 东南亚各国获得信贷便利度
- 印尼获得信贷便利度
- 总结：印尼私人信贷市场还有很大发展空间

### 2.2.1 画图 ( Politics )

```
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

def create_bar_chart(latest_data_sorted, latest_year):
 """ 创建各国私人信贷占 GDP 比重对比的柱状图 """

 countries_list = latest_data_sorted['Country'].tolist()
 values_list = latest_data_sorted['Private_Credit_GDP'].tolist()

 # 图表1：柱状图
 fig = px.bar(
 latest_data_sorted,
 y='Country',
 x='Private_Credit_GDP',
```

```

 orientation='h',
 text=[f'{v:.1f}% for v in values_list],
 title=f'{latest_year}年各国私人信贷占GDP比重对比',
 labels={'Private_Credit_GDP': '私人信贷占GDP比重 (%)',
'Country': '国家'},
 color='Country',
 color_discrete_map={'Indonesia': 'red', 'Philippines': 'blue'}
印尼用红色
)

fig.update_traces(textposition='outside')
fig.update_layout(showlegend=False)

调整图表尺寸，使其适合Colab打印展示
fig.update_layout(
 height=400, # 设置合适的高度
 width=700, # 设置合适的宽度
 margin=dict(l=50, r=100, t=80, b=100), # 右侧增加空间，修复泰国%
被切掉的bug
 title_font=dict(size=16), # 调整标题大小
 font=dict(size=11), # 调整整体字体大小
 xaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10),
 range=[0, max(values_list) * 1.15] # 调整X轴范围，给文字留出
空间
),
 yaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10),
 automargin=True, # 自动调整边距确保国家名称完全显示
 categoryorder='total ascending' # 按数值升序排列
)
)

调整柱状图文字大小和位置
fig.update_traces(
 textfont=dict(size=9),
 textposition='outside',
 textangle=0
)

fig.show()

def create_line_chart(indonesia_data):
 """创建印尼私人信贷占GDP比重趋势的折线图"""

 years_list = indonesia_data['Year'].astype(str).tolist()
 indo_values = indonesia_data['Private_Credit_GDP'].tolist()

```

```

图表2：折线图
fig = px.line(
 indonesia_data,
 x='Year',
 y='Private_Credit_GDP',
 markers=True,
 title=f'印尼私人信贷占GDP比重趋势 ({years_list[0]}-{years_list[-1]})',
 labels={'Private_Credit_GDP': '私人信贷占GDP比重 (%)', 'Year': '年份'}
)

fig.update_traces(
 line=dict(color='red', width=2.5),
 marker=dict(size=8, color='white', line=dict(color='red', width=2))
)

添加数据标签
for i, (year, value) in enumerate(zip(years_list, indo_values)):
 fig.add_annotation(
 x=year,
 y=value,
 text=f'{value:.1f}%',
 showarrow=False,
 yshift=10,
 font=dict(size=10, color='black')
)

调整图表尺寸
fig.update_layout(
 height=400, # 设置合适的高度
 width=700, # 设置合适的宽度
 margin=dict(l=50, r=50, t=80, b=80), # 调整边距
 title_font=dict(size=16), # 调整标题大小
 font=dict(size=11), # 调整整体字体大小
 xaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10),
 tickmode='array',
 tickvals=years_list
),
 yaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10),
 range=[min(indo_values) * 0.9, max(indo_values) * 1.1] # 调整Y轴范围
)
)

```

```

fig.show()

import requests

print("获取私人信贷占 GDP 比重时间序列数据 (2015-2024)...")
print("*"*60)

定义目标国家和指标
countries = {
 'ID': 'Indonesia',
 'MY': 'Malaysia',
 'SG': 'Singapore',
 'TH': 'Thailand',
 'PH': 'Philippines'
}
country_codes = ';' .join(countries.keys())
indicator = 'FS.AST.PRVT.GD.ZS' # 私人信贷占GDP 比重

构建API URL - 获取2015-2025 年数据
url =
f"https://api.worldbank.org/v2/country/{country_codes}/indicator/{indicator}?format=json&date=2015:2025&per_page=1000"
response = requests.get(url)
data = response.json()

提取并整理数据
if len(data) > 1:
 records = []
 for item in data[1]:
 country_code = item['countryiso3code']
 country_name = countries.get(country_code, item['country']['value'])
 records.append({
 'Country': country_name,
 'Country_Code': country_code,
 'Year': int(item['date']),
 'Private_Credit_GDP': item['value']
 })
df = pd.DataFrame(records)

按国家和年份排序
df = df.sort_values(['Country', 'Year']).reset_index(drop=True)

print("□ 数据获取成功！")
print(f"数据形状: {df.shape}")
print(f"年份范围: {df['Year'].min()} - {df['Year'].max()}")
print("\n□ 数据预览:")
print(df.head(10))

```

```

else:
 print("□ 无法获取数据")

获取私人信贷占 GDP 比重时间序列数据 (2015-2024)...
=====
□ 数据获取成功 !
数据形状: (50, 4)
年份范围: 2015 - 2024

□ 数据预览:
 Country Country_Code Year Private_Credit_GDP
0 Indonesia IDN 2015 39.118798
1 Indonesia IDN 2016 39.402425
2 Indonesia IDN 2017 38.732350
3 Indonesia IDN 2018 38.805450
4 Indonesia IDN 2019 38.372694
5 Indonesia IDN 2020 38.679539
6 Indonesia IDN 2021 36.990827
7 Indonesia IDN 2022 35.288433
8 Indonesia IDN 2023 36.012889
9 Indonesia IDN 2024 36.392451

找到所有国家都有数据的最近年份
def find_latest_common_year(df):
 """找到所有5个国家都有数据的最近年份"""
 # 获取每个国家有数据的年份
 country_years = {}
 for country in df['Country'].unique():
 country_data = df[df['Country'] == country].dropna(subset=['Private_Credit_GDP'])
 if not country_data.empty:
 country_years[country] = set(country_data['Year'])

 # 找到所有国家都有的年份
 common_years = set.intersection(*country_years.values())
 if common_years:
 return max(common_years)
 return None

latest_year = find_latest_common_year(df)

if latest_year:
 #print(f"\n□ 最新共同年份: {latest_year} 年(所有5个国家都有数据)")

 # 提取最近年份数据
 latest_data = df[df['Year'] == latest_year].dropna(subset=['Private_Credit_GDP'])
 # print(f"\n□ {latest_year} 年各国数据:")
 # print("-" * 40)
 # for _, row in latest_data.iterrows():

```

```

print(f"{row['Country']}: {row['Private_Credit_GDP']:.1f}%")

创建可视化图表
create_bar_chart(latest_data, latest_year)
else:
 print("\n⚠ 无法找到所有国家都有数据的共同年份")

查看近 10 年印度尼西亚私人信贷占 GDP 比重的趋势变化
indonesia_data = df[df['Country'] ==
'Indonesia'].dropna(subset=['Private_Credit_GDP']).sort_values('Year')
create_line_chart(indonesia_data)

```

## 2.2.2 信贷制度洞察：印尼私人信贷市场还有很大发展空间

关键发现：印尼的私人信贷规模明显落后于邻国

数据对比（2020 年）：

- 印尼：38.7%
- 菲律宾：52.0%（比印尼高 13 个百分点）
- 马来西亚：133.8%（约是印尼的 3.5 倍）
- 新加坡：129.2%（约是印尼的 3.3 倍）
- 泰国：160.1%（约是印尼的 4 倍）

这意味着什么？

1. 市场不发达：印尼的私人信贷（包括信用卡、个人贷款等）规模还很小
2. 增长空间大：相比邻国，印尼的信贷市场还有 2-3 倍的增长潜力
3. 机会明确：Yup 所处的市场远未饱和

对 Yup 的启示：

机会：市场有巨大发展空间，远未到天花板 挑战：需要教育市场，提升信贷接受度 定位：Yup 不仅是服务“被拒绝者”，更是在推动&教育整个印尼市场的增长

## 2.3 东南亚互联网普率对比及印尼趋势

### 2.3.1 画图（Tech）

```

import pandas as pd
import requests
import plotly.graph_objects as go
from plotly.subplots import make_subplots

使用世界银行 API 获取数据
def get_worldbank_data(indicator_code, country_code='ID',
start_year=2015, end_year=2025):
 """
 获取世界银行数据
 indicator_code: 指标代码

```

```

country_code: 国家代码 (ID=印尼, MY=马来西亚等)
"""
url =
f"https://api.worldbank.org/v2/country/{country_code}/indicator/{indicator_code}"
params = {
 'format': 'json',
 'date': f'{start_year}:{end_year}',
 'per_page': 100
}
response = requests.get(url, params=params)
data = response.json()

提取数据
records = []
if len(data) > 1:
 for item in data[1]:
 records.append({
 'year': item['date'],
 'value': item['value'],
 'country': item['country']['value']
 })
df = pd.DataFrame(records)
df['value'] = pd.to_numeric(df['value'], errors='coerce')
return df

获取互联网普及率数据
指标代码 : IT.NET.USER.ZS
print("正在获取印尼互联网普及率数据...")
df_internet_id = get_worldbank_data('IT.NET.USER.ZS', 'ID', 2015, 2023)

print("正在获取对比国家数据...")
获取其他国家数据用于对比
countries = {
 '马来西亚': 'MY',
 '泰国': 'TH',
 '菲律宾': 'PH',
 '新加坡': 'SG'
}

all_data = {'印尼': df_internet_id}

for country_name, country_code in countries.items():
 df_temp = get_worldbank_data('IT.NET.USER.ZS', country_code, 2015, 2023)
 all_data[country_name] = df_temp

```

```

print("数据获取完成！")
print("\n 印尼数据预览:")
print(df_internet_id.head())

正在获取印尼互联网普及率数据...
正在获取对比国家数据...
数据获取完成！

印尼数据预览:
 year value country
0 2023 69.2084 Indonesia
1 2022 66.4846 Indonesia
2 2021 62.1045 Indonesia
3 2020 53.7265 Indonesia
4 2019 47.6906 Indonesia

1. 准备数据：整理成统一格式用于绘图
将各国数据合并成一个数据框，按年份和国别组织
data_for_plot = []

for country_name, df_country in all_data.items():
 if not df_country.empty:
 # 确保数据按年份排序
 df_sorted = df_country.sort_values('year')
 for _, row in df_sorted.iterrows():
 if pd.notna(row['value']):
 data_for_plot.append({
 'country': country_name,
 'year': row['year'],
 'value': row['value']
 })
df_plot = pd.DataFrame(data_for_plot)

2. 创建包含1行、2列的子图框架 - 增加水平间距
fig = make_subplots(
 rows=1, cols=2,
 subplot_titles=(
 "印尼互联网普及率增长趋势",
 "2023年东南亚各国互联网普及率对比"
),
 shared_yaxes=True, # 共享Y轴，便于对比数值
 horizontal_spacing=0.3 # 进一步增加两个图之间的水平间距，避免重叠
)

3. 为左图（第一列）添加线条：印尼时间趋势
提取印尼数据
indo_data = df_plot[df_plot['country'] == '印尼'].sort_values('year')

fig.add_trace(

```

```

go.Scatter(
 x=indo_data['year'],
 y=indo_data['value'],
 mode='lines+markers',
 name='印尼',
 line=dict(color='red', width=2.5), # 稍微减小线宽
 marker=dict(size=8, symbol='circle'), # 稍微减小标记大小
 hovertemplate='年份: %{x}
互联网普及率: %
{y:.1f}'%<extra></extra>',
),
 row=1, col=1
)

4. 为右图(第二列)添加条形图:2023年各国对比 - 修改国家名字显示方式
提取2023年数据
data_2023 = []
for country_name, df_country in all_data.items():
 if not df_country.empty:
 # 查找2023年数据
 data_2023_country = df_country[df_country['year'] == '2023']
 if not data_2023_country.empty:
 value = data_2023_country['value'].iloc[0]
 if pd.notna(value):
 data_2023.append({
 'country': country_name,
 'value': value
 })

转换为DataFrame并排序
if data_2023:
 df_2023 = pd.DataFrame(data_2023)
 df_2023 = df_2023.sort_values('value', ascending=True)

创建颜色列表(印尼用红色突出)
colors = []
for country in df_2023['country']:
 if country == '印尼':
 colors.append('red')
 elif country == '新加坡':
 colors.append('green')
 elif country == '马来西亚':
 colors.append('orange')
 elif country == '泰国':
 colors.append('purple')
 elif country == '菲律宾':
 colors.append('blue')
 else:
 colors.append('gray')

```

```

添加条形图 - 修改这里：国家名字和数据显示在柱状图上
fig.add_trace(
 go.Bar(
 x=df_2023['value'],
 y=df_2023['country'],
 orientation='h',
 marker=dict(color=colors),
 # 修改这里：在柱状图上显示国家名字和数值
 text=[f'{country}
{value:.1f}%' for country, value in
zip(df_2023['country'], df_2023['value'])],
 textposition='inside', # 改为inside，让文字显示在柱状图内部
 insidetextanchor='start', # 文字从柱状图的开始位置显示
 textfont=dict(
 size=9,
 color='white', # 设置文字颜色为白色，提高对比度
 weight='bold' # 加粗字体
),
 # 移除原来的hovertemplate，因为信息已经在柱状图上了
 name='印尼 2023 年数据'
),
 row=1, col=2
)

5. 更新整体布局 - 调整尺寸为适合 Colab 展示
fig.update_layout(
 hovermode='closest',
 height=400, # 减小高度，避免被裁剪
 width=900, # 减小宽度，适合 Colab 展示
 showlegend=True,
 legend=dict(
 orientation="h",
 yanchor="bottom",
 y=1.02,
 xanchor="center",
 x=0.5,
 font=dict(size=10) # 稍微减小图例字体
),
 margin=dict(l=50, r=50, t=80, b=80), # 增加底部边距，避免文字被裁剪
 font=dict(size=10) # 整体减小字体大小
)

6. 更新 X 轴和 Y 轴标签 - 调整标题位置
fig.update_xaxes(
 title_text="年份",
 title_font=dict(size=11), # 稍微减小标题字体
 row=1, col=1,
 tickfont=dict(size=9) # 稍微减小刻度字体
)
fig.update_xaxes(

```

```

 title_text="互联网普及率 (%)",
 title_font=dict(size=11), # 稍微减小标题字体
 row=1, col=2,
 tickfont=dict(size=9) # 稍微减小刻度字体
)
fig.update_yaxes(
 title_text="互联网普及率 (%)",
 title_font=dict(size=11), # 稍微减小标题字体
 row=1, col=1,
 tickfont=dict(size=9) # 稍微减小刻度字体
)
fig.update_yaxes(
 title_text="国家",
 title_font=dict(size=11), # 稍微减小标题字体
 row=1, col=2,
 tickfont=dict(size=9) # 稍微减小刻度字体
)

更新子图标题位置，使其与图表有适当距离
fig.update_annotations(
 dict(
 font_size=13, # 稍微减小子图标题字体
 font_weight="bold",
 yshift=15 # 稍微减小标题上移距离
)
)

7. 显示图表
fig.show()

8. 打印关键数据洞察
print("\n" + "*60)
print("数据关键洞察:")
print("*60)

印尼趋势分析
if not indo_data.empty:
 # 计算增长率
 if len(indo_data) >= 2:
 first_year = indo_data['year'].iloc[0]
 first_value = indo_data['value'].iloc[0]
 last_year = indo_data['year'].iloc[-1]
 last_value = indo_data['value'].iloc[-1]

 total_growth = last_value - first_value
 years_diff = int(last_year) - int(first_year)
 annual_growth = total_growth / years_diff if years_diff > 0
else 0

 print(f"1. 印尼互联网增长趋势 ({first_year}-{last_year}):")

```

```

print(f" • 起始水平: {first_value:.1f}%")
print(f" • 当前水平: {last_value:.1f}%")
print(f" • 总增长: +{total_growth:.1f}个百分点")
print(f" • 年均增长率: +{annual_growth:.1f}个百分点/年")

2023 年对比分析
if data_2023:
 df_2023_sorted = pd.DataFrame(data_2023).sort_values('value',
ascending=False)

 print(f"\n2. 2023 年各国对比 (共{len(df_2023_sorted)}个国家):")

 # 找到印尼的位置
 if '印尼' in df_2023_sorted['country'].values:
 indo_value = df_2023_sorted[df_2023_sorted['country'] == '印
尼']['value'].values[0]
 indo_rank = list(df_2023_sorted['country']).index('印尼') + 1

 # 找到最高和最低
 top_country = df_2023_sorted.iloc[0]['country']
 top_value = df_2023_sorted.iloc[0]['value']

 print(f" • 印尼: {indo_value:.1f}% (排名第{indo_rank}位)")
 print(f" • 最高: {top_country} ({top_value:.1f}%)")
 print(f" • 与最高差距: {top_value - indo_value:.1f}个百分点")

 # 计算区域平均 (不含新加坡, 因为太特殊)
 other_countries = ['马来西亚', '泰国', '菲律宾']
 other_values = []
 for country in other_countries:
 if country in df_2023_sorted['country'].values:
 other_values.append(df_2023_sorted[df_2023_sorted['country'] ==
country]['value'].values[0])

 if other_values:
 regional_avg = sum(other_values) / len(other_values)
 print(f" • 区域平均 (马来西亚/泰国/菲律宾): {regional_avg:.1f}%")
 print(f" • 印尼与区域平均对比: {indo_value - regional_avg:.1f}%")

print("\n3. 对金融科技 (如 YUP 信用卡) 的启示:")
print(" • 互联网普及率提供了数字基础设施")
print(" • 互联网接入与金融接入之间的差距 = 市场机会")
print(" • 印尼的快速增长支持数字化优先策略")
print("=="*60)

补充: 显示原始数据表格
print("\n原始数据汇总:")

```

```
print("-"*40)
for country_name, df_country in all_data.items():
if not df_country.empty:
print(f"\n{country_name}:")
df_display = df_country.sort_values('year')
for _, row in df_display.iterrows():
if pd.notna(row['value']):
print(f" {row['year']} 年: {row['value']:.1f}%")
```

#### ===== 数据关键洞察：

##### 1. 印尼互联网增长趋势 (2015-2023) :

- 起始水平: 22.1%
- 当前水平: 69.2%
- 总增长: +47.1 个百分点
- 年均增长率: +5.9 个百分点/年

##### 2. 2023 年各国对比 (共 5 个国家) :

- 印尼: 69.2% (排名第 5 位)
- 最高: 马来西亚 (97.7%)
- 与最高差距: 28.5 个百分点
- 区域平均 (马来西亚/泰国/菲律宾): 90.3%
- 印尼与区域平均对比: -21.1%

##### 3. 对金融科技 (如 YUP 信用卡) 的启示:

- 互联网普及率提供了数字基础设施
- 互联网接入与金融接入之间的差距 = 市场机会
- 印尼的快速增长支持数字化优先策略

### 2.3.2 技术基础洞察：印尼互联网普及率快速增长，存在线上金融缺口

核心发现：印尼互联网普及率 9 年间从 22% 快速增长到 69%，虽然仍落后于邻国，但数字基础已基本建成。

对 **Yup** 的意义：

1. 可行性验证：足够多的人上网 → 纯线上服务模式可行
2. 增长动力：快速普及趋势 → 未来会有更多“数字原住民”
3. 机会窗口：互联网(69%) > 银行账户(约 65%) → 印尼人现在缺的是好的线上金融服务，这正是 **Yup** 的机会

## 2.4 收入结构与市场细分

### 2.4.1 画图 (Economics)

```
import pandas as pd
import requests
import plotly.graph_objects as go
from plotly.subplots import make_subplots

使用世界银行API 获取数据
def get_worldbank_data(indicator_code, country_code='ID',
start_year=2015, end_year=2025):
 """
 获取世界银行数据
 indicator_code: 指标代码
 country_code: 国家代码 (ID=印尼, MY=马来西亚等)
 """
 url =
f"https://api.worldbank.org/v2/country/{country_code}/indicator/{indicator_code}"
 params = {
 'format': 'json',
 'date': f'{start_year}:{end_year}',
 'per_page': 100
 }

 response = requests.get(url, params=params)
 data = response.json()

 # 提取数据
 records = []
 if len(data) > 1:
 for item in data[1]:
 records.append({
 'year': item['date'],
 'value': item['value'],
 'country': item['country']['value']
 })

 df = pd.DataFrame(records)
 df['value'] = pd.to_numeric(df['value'], errors='coerce')
 return df

print("正在获取印尼人均GDP数据...")
获取人均GDP数据(指标代码:NY.GDP.PCAP.CD)
df_gdp_id = get_worldbank_data('NY.GDP.PCAP.CD', 'ID', 2015, 2023)

print("正在获取印尼基尼系数数据...")
获取基尼系数数据(指标代码:SI.POV.GINI)
df_gini_id = get_worldbank_data('SI.POV.GINI', 'ID', 2015, 2023)
```

```
print("数据获取完成！")
print("\n印尼人均 GDP 数据预览:")
print(df_gdp_id.head())
print("\n印尼基尼系数数据预览:")
print(df_gini_id.head())

正在获取印尼人均 GDP 数据...
正在获取印尼基尼系数数据...
数据获取完成！

印尼人均 GDP 数据预览:
 year value country
0 2023 4876.307745 Indonesia
1 2022 4730.834848 Indonesia
2 2021 4287.173140 Indonesia
3 2020 3853.702888 Indonesia
4 2019 4106.948562 Indonesia

印尼基尼系数数据预览:
 year value country
0 2023 36.1 Indonesia
1 2022 35.5 Indonesia
2 2021 35.5 Indonesia
3 2020 35.3 Indonesia
4 2019 35.4 Indonesia

1. 创建简单的双图表布局
fig = make_subplots(
 rows=1, cols=2,
 subplot_titles=(
 "印尼人均 GDP 增长趋势",
 "印尼收入不平等程度（基尼系数）"
),
 horizontal_spacing=0.25
)

2. 图表1：人均GDP趋势（简单柱状图）
if not df_gdp_id.empty:
 # 清理数据，只保留有数据的年份
 df_gdp_clean =
df_gdp_id.dropna(subset=['value']).sort_values('year')

 # 计算年增长率
 df_gdp_clean['growth_rate'] = df_gdp_clean['value'].pct_change() *
100

 # 添加柱状图
 fig.add_trace(
 go.Bar(
```

```

 x=df_gdp_clean['year'],
 y=df_gdp_clean['value'],
 name='人均GDP(美元)',
 marker_color='lightblue',
 hovertemplate='年份: %{x}
人均GDP:
$%{y:.0f}<extra></extra>',
 text=[f'${v:.0f}' for v in df_gdp_clean['value']],
 textposition='outside'
),
 row=1, col=1
)

添加趋势线
fig.add_trace(
 go.Scatter(
 x=df_gdp_clean['year'],
 y=df_gdp_clean['value'],
 mode='lines',
 name='趋势线',
 line=dict(color='red', width=2),
 showlegend=False,
 hovertemplate='年份: %{x}
人均GDP:
$%{y:.0f}<extra></extra>'
),
 row=1, col=1
)

3. 图表2：基尼系数趋势（简单折线图）
if not df_gini_id.empty:
 # 清理数据，只保留有数据的年份
 df_gini_clean =
df_gini_id.dropna(subset=['value']).sort_values('year')

 # 添加折线图
 fig.add_trace(
 go.Scatter(
 x=df_gini_clean['year'],
 y=df_gini_clean['value'],
 mode='lines+markers',
 name='基尼系数',
 line=dict(color='orange', width=3),
 marker=dict(size=8),
 hovertemplate='年份: %{x}
基尼系数: %
{y:.1f}<extra></extra>',
 text=[f'{v:.1f}' for v in df_gini_clean['value']],
 textposition='top center'
),
 row=1, col=2
)

```

```

添加参考线：基尼系数警戒线（通常 0.4 以上表示收入差距较大）
fig.add_hline(
 y=40,
 line_dash="dash",
 line_color="red",
 opacity=0.5,
 row=1, col=2,
 annotation_text="警戒线 (0.4)",
 annotation_position="top right"
)

4. 更新布局
fig.update_layout(
 title_text='印尼经济水平与收入分配分析',
 height=500,
 width=1000,
 showlegend=True,
 legend=dict(
 orientation="h",
 yanchor="bottom",
 y=1.02,
 xanchor="center",
 x=0.5
),
 hovermode='x unified'
)

5. 更新坐标轴标签
fig.update_xaxes(title_text="年份", row=1, col=1)
fig.update_xaxes(title_text="年份", row=1, col=2)
fig.update_yaxes(title_text="人均 GDP (美元)", row=1, col=1)
fig.update_yaxes(title_text="基尼系数", row=1, col=2)

6. 显示图表
fig.show()

7. 简单分析：中低收入群体规模估算
print("\n" + "*60")
print("印尼中低收入群体规模分析")
print("*60")

如果有基尼系数，可以简单估算
if not df_gini_id.empty and not df_gdp_id.empty:
 # 获取最新数据
 latest_gdp =
df_gdp_id.dropna(subset=['value']).sort_values('year').iloc[-1]
['value']
 latest_gini =
df_gini_id.dropna(subset=['value']).sort_values('year').iloc[-1]

```

```

['value']
latest_gdp_year =
df_gdp_id.dropna(subset=['value']).sort_values('year').iloc[-1]
['year']

print(f"1. 最新数据 ({latest_gdp_year}年):")
print(f" • 人均 GDP: ${latest_gdp:.0f}")
print(f" • 基尼系数: {latest_gini:.1f}")

简单估算中低收入群体比例 (基于国际经验)
基尼系数越高，中低收入群体占比越大
if latest_gini < 30:
 middle_low_ratio = 0.4 # 40%
elif latest_gini < 35:
 middle_low_ratio = 0.5 # 50%
elif latest_gini < 40:
 middle_low_ratio = 0.6 # 60%
else:
 middle_low_ratio = 0.7 # 70%或以上

print(f"\n2. 中低收入群体估算:")
print(f" • 基尼系数 {latest_gini:.1f} 表明收入不平等程度较高")
print(f" • 估算中低收入群体占比: {middle_low_ratio*100:.0f}%")
print(f" • 主要特征: 收入有限但稳定, 有消费和信贷需求")

print(f"\n3. 对 YUP 信用卡的启示:")
print(f" • 目标市场: 约占人口{middle_low_ratio*100:.0f}%的中低收入群
体")
 print(f" • 收入水平: 人均约${latest_gdp*0.3:.0f}-$ {latest_gdp*0.8:.0f}/年")
 print(f" • 金融需求: 小额信贷、消费分期、基础银行服务")
else:
 print("数据不足, 无法进行详细分析")

print("=*60")

```

=====  
印尼中低收入群体规模分析  
=====

1. 最新数据 (2023 年):
  - 人均 GDP: \$4,876
  - 基尼系数: 36.1
2. 中低收入群体估算:
  - 基尼系数 36.1 表明收入不平等程度较高
  - 估算中低收入群体占比: 60%
  - 主要特征: 收入有限但稳定, 有消费和信贷需求
3. 对 YUP 信用卡的启示:

- 目标市场：约占人口 60%的中低收入群体
  - 收入水平：人均约\$1,463-\$3,901/年
  - 金融需求：小额信贷、消费分期、基础银行服务
- 

## 2.4.2 经济环境洞察：庞大的中低收入市场

核心发现：印尼 60% 人口属于中低收入群体（年收入约 1,500-3,900 美刀），人数多但服务不足。

对 **Yup** 的意义：

1. 市场定位：正好服务主流人群（60% 人口），不是小众市场
2. 产品匹配：适合小额信贷、分期消费等产品
3. 支付能力：收入足够支撑信用卡消费，但需要合理额度

Yup 服务的是印尼的“大多数人”——他们有稳定收入但传统银行看不上，这正是最大机会所在。

# 三：印尼数字信贷市场用户分析：竞争格局与 Yup 机会

## 用户研究：基于 Google Trends 五年搜索数据的趋势演变、竞争对比与空白市场识别

## 3.1 验证基础需求

搜索词：kartu kredit (信用卡) vs pinjaman online (在线贷款)

---

分析潜在用户的市场基本盘：印尼用户对传统产品和数字产品的关注度趋势如何？数字产品是否在侵蚀传统市场？

### 3.1.1 画图

```
1. 导入必要的库
import pandas as pd
import plotly.express as px
from google.colab import files

2. 上传你在 Google Trends 下载的 CSV 文件到 Colab
上传'multiTimeline (1).csv'
print("请上传你从 Google Trends 下载的 CSV 文件：")
uploaded = files.upload()
```

请上传你从 Google Trends 下载的 CSV 文件：

```
<IPython.core.display.HTML object>
```

```
Saving multiTimeline (1).csv to multiTimeline (1).csv
```

```
3. 读取数据(注意:Google Trends 的CSV文件开头有几行描述,需要跳过)
获取上传的文件名
file_name = list(uploaded.keys())[0]
跳过前2行(通常是表头描述),并正确设置列名
df = pd.read_csv(file_name, skiprows=2)
查看前几行,了解数据结构
print("数据前5行:")
print(df.head())
print("\n数据列名:")
print(df.columns)
```

数据前5行:

|   | Week       | kartu kredit: (Indonesia) | pinjaman online: (Indonesia) |
|---|------------|---------------------------|------------------------------|
| 0 | 2020-12-27 | 90                        | 76                           |
| 1 | 2021-01-03 | 87                        | 78                           |
| 2 | 2021-01-10 | 84                        | 75                           |
| 3 | 2021-01-17 | 89                        | 78                           |
| 4 | 2021-01-24 | 86                        | 78                           |

pinjaman tanpa kartu kredit: (Indonesia) pinjaman cepat online: (Indonesia)

|   |   |
|---|---|
| 0 | 1 |
| 7 |   |
| 1 | 1 |
| 8 |   |
| 2 | 1 |
| 7 |   |
| 3 | 1 |
| 7 |   |
| 4 | 1 |
| 8 |   |

数据列名:

```
Index(['Week', 'kartu kredit: (Indonesia)', 'pinjaman online: (Indonesia)',
 'pinjaman tanpa kartu kredit: (Indonesia)',
 'pinjaman cepat online: (Indonesia)'],
 dtype='object')
```

# 4. 数据清洗与格式化(这是最关键的一步)

```
4.1 重命名列,使其更易读(你的列名可能略有不同,请根据上一步的打印结果调整)
df = df.rename(columns={
```

```

'Week': 'date', # 日期列
'kartu kredit: (Indonesia)': 'credit_card_interest', # 信用卡兴趣值
'pinjaman online: (Indonesia)': 'online_loan_interest', # 在线贷款兴趣值
'pinjaman tanpa kartu kredit: (Indonesia)': \
'loan_without_credit_card_interest', # 无信用卡贷款值
'pinjaman cepat online: (Indonesia)': 'fast_online_loan_interest'
快速在线贷款值
})

4.2 确保日期是日期格式
df['date'] = pd.to_datetime(df['date'])
4.3 检查缺失值
print(f"\n 数据缺失情况 : \n{df.isnull().sum()}")
4.4 处理缺失值 : 用前后值的平均值填充
df = df.fillna()
保存处理好的数据，供后续分析使用
df.to_csv('processed_google_trends_data.csv', index=False)
print("\n 数据处理完成，已保存为 'processed_google_trends_data.csv'。")

```

数据缺失情况：

|                                   |   |
|-----------------------------------|---|
| date                              | 0 |
| credit_card_interest              | 0 |
| online_loan_interest              | 0 |
| loan_without_credit_card_interest | 0 |
| fast_online_loan_interest         | 0 |
| dtype: int64                      |   |

数据处理完成，已保存为 'processed\_google\_trends\_data.csv'。

```

5. 制作一个简单的趋势对比图
fig = px.line(df,
 x='date',
 y=['credit_card_interest', 'online_loan_interest'],
 title='印尼市场对“信用卡”与“在线贷款”的搜索兴趣对比 (过去 5 年)',
 labels={
 'value': '相对搜索兴趣值',
 'date': '日期',
 'variable': '搜索关键词'
 },
 color_discrete_map={
 'credit_card_interest': 'blue',
 'online_loan_interest': 'red'
 })
fig.update_layout(hovermode='x unified')
在显示前调整图表尺寸，使其适合 Colab 打印展示
fig.update_layout(
 height=400, # 设置合适的高度
 width=800, # 设置合适的宽度
)

```

```

margin=dict(l=50, r=50, t=80, b=80), # 调整边距
title_font=dict(size=16), # 调整标题大小
font=dict(size=11), # 调整整体字体大小
xaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10)
),
yaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10)
),
legend=dict(
 title_font=dict(size=11),
 font=dict(size=10),
 yanchor="top",
 y=0.99,
 xanchor="left",
 x=0.01,
 bgcolor='rgba(255, 255, 255, 0.8)'
)
)

调整线条样式，使其更清晰
fig.update_traces(
 line=dict(width=2),
 marker=dict(size=4)
)

fig.show()

```

### 3.1.2 用户基本盘：数字借贷需求存在波动，传统信用卡需求稳健

根据过去五年的搜索趋势数据，可以观察到两个关键现象：

1. 趋势对比：自 2021 年起，用户对“线上贷款”的搜索兴趣与对“传统信用卡”的兴趣出现短期趋同，表明数字金融产品在当时获得了高度关注。然而，2022 年后，“线上贷款”的搜索热度呈现显著且持续的下跌趋势，而“传统信用卡”的搜索兴趣始终保持在一个相对平稳的区间内。
2. 初步结论：数据表明，在印尼市场，数字借贷产品尚未能持续侵蚀或替代传统信用卡的市场关注度。传统信用卡的需求展现出更强的稳定性和韧性。

### 3.1.3 对本研究项目的启示

线上贷款搜索量下降，说明：

- 市场在“试错”后变得更理性
- 机会不在跟风做同类产品
- 而是做靠谱的数字化方案

YUP 正好卡在这个位置：

- 比传统信用卡更方便（数字体验）
- 比线上贷款更可靠（visa 卡，正规产品）

结论：大家都在找“靠谱又方便”的信贷产品，YUP 这样旨在弥合传统金融与数字鸿沟的创新产品有机会填补这个空白。

## 3.2 分析现有竞争格局

搜索词：Akulaku vs Kredivo vs Shopee Paylater

---

头部玩家对比：在“先买后付”这个成熟赛道里，谁更占优势？用户心智被谁占据？

### 3.2.1 画图

```
1 上传你在 Google Trends 下载的 CSV 文件 2 到 Colab

上传'multiTimeline (2).csv'
print("请上传你从 Google Trends 下载的 CSV 文件：")
uploaded2 = files.upload()

请上传你从 Google Trends 下载的 CSV 文件：

<IPython.core.display.HTML object>

Saving multiTimeline (2).csv to multiTimeline (2).csv

2 读取数据
file_name2 = list(uploaded2.keys())[0]
跳过前 2 行（通常是表头描述），并正确设置列名
df2 = pd.read_csv(file_name2, skiprows=2)
查看前几行，了解数据结构
print("数据前 5 行：")
print(df2.head())
print("\n 数据列名：")
print(df2.columns)
6.3. 数据清洗与格式化
df2 = df2.rename(columns={
 'Week': 'date', # 日期列
 'yup: (Indonesia)': 'yup_interest', # yup 兴趣值
 'Akulaku: (Indonesia)': 'akulaku_interest', # akulaku 兴趣值
 'kartu kredit: (Indonesia)': 'credit_card_interest', # 信用卡兴趣值
 'Shopee Paylater: (Indonesia)': 'shopee_paylater_interest',
 # shopee 先用后付兴趣值
 'Kredivo: (Indonesia)': 'kredivo_interest' # kredivo 兴趣值
})

df2['date'] = pd.to_datetime(df2['date'])
print(f"\n 数据缺失情况：\n{df2.isnull().sum()}"")
```

```

df2 = df2.fillna(0)
保存处理好的数据，供后续分析使用
df2.to_csv('processed_google_trends_data2.csv', index=False)
print("\n 数据处理完成，已保存为 'processed_google_trends_data2.csv'。")

```

数据缺失情况：

|                          | 0 |
|--------------------------|---|
| date                     | 0 |
| yup_interest             | 0 |
| akulaku_interest         | 0 |
| credit_card_interest     | 0 |
| shopee_paylater_interest | 0 |
| kredivo_interest         | 0 |

dtype: int64

数据处理完成，已保存为 'processed\_google\_trends\_data2.csv'。

```

====== 诊断与修复代码块START ======
print("== 诊断开始==")
print("1. 当前数据框 df2 的所有列名:")
print(df2.columns.tolist())

2. 自动识别可能的数值列（根据你的命名，假设以 '_interest' 结尾的列都是兴趣值）
如果列名规则不同，你需要手动指定 interest_columns 列表
interest_columns = ['yup_interest', 'akulaku_interest',
'credit_card_interest', 'shopee_paylater_interest',
'kredivo_interest']

print(f"\n2. 计划处理的兴趣值列: {interest_columns}")
print("\n3. 处理前，这些列的数据类型:")
print(df2[interest_columns].dtypes)

3. 关键修复：循环处理每一列，强制转换为数值类型，非数字变为NaN
for col in interest_columns:
 # pd.to_numeric 是核心，errors='coerce' 会将无法转换的值（如空字符串、'<1'）设为NaN
 df2[col] = pd.to_numeric(df2[col], errors='coerce')

print("\n4. 转换数值类型后，这些列的数据类型:")
print(df2[interest_columns].dtypes)

4. 处理转换产生的缺失值 (NaN)：用前向填充（用前面的有效值填充后面的NaN）
df2[interest_columns] = df2[interest_columns].ffill()
如果开头还有NaN，再用后向填充一次
df2[interest_columns] = df2[interest_columns].ffill()

print("\n5. 处理后，数据前几行（查看填充效果）:")
print(df2[['date'] + interest_columns].head())

print("\n== 诊断与修复完成，可以绘图了==")

```

```
制作两个独立的趋势对比图
import plotly.graph_objects as go
from plotly.subplots import make_subplots

1. 创建第一个图表：分析现有竞争格局
fig1 = go.Figure()

添加线条：分析现有竞争格局
对比Akulaku, Kredivo, Shopee Paylater
fig1.add_trace(
 go.Scatter(
 x=df2['date'],
 y=df2['akulaku_interest'],
 mode='lines',
 name='Akulaku',
 line=dict(color='red'),
 hovertemplate='日期: %{x}
兴趣值: %{y}<extra></extra>'
)
)
fig1.add_trace(
 go.Scatter(
 x=df2['date'],
 y=df2['kredivo_interest'],
 mode='lines',
 name='Kredivo',
 line=dict(color='purple'),
 hovertemplate='日期: %{x}
兴趣值: %{y}<extra></extra>'
)
)
fig1.add_trace(
 go.Scatter(
 x=df2['date'],
 y=df2['shopee_paylater_interest'],
 mode='lines',
 name='Shopee Paylater',
 line=dict(color='orange'),
 hovertemplate='日期: %{x}
兴趣值: %{y}<extra></extra>'
)
)

更新第一个图表的布局
fig1.update_layout(
 title='印尼数字信贷市场：头部 BNPL 玩家对比 (2021-2025)',
 hovermode='x unified',
 height=500,
 width=600,
 xaxis_title="日期",
 yaxis_title="相对搜索兴趣值"
)
```

```
3. 显示两个独立的图表
fig1.show()
```

### 3.2.2 印尼三大“先买后付”平台对比

通过对比印尼三大“先买后付”平台过去五年的搜索兴趣趋势，可以清晰地看到市场格局的演变：

格局变迁：市场从 2021 年“一超多强”（Akulaku 显著领先，Kredivo 与 Shopee Paylater 紧随其后）的格局，演变为目前“双雄竞争，一者掉队”的局面。

趋势变化：

- **Akulaku**：搜索兴趣从 2021 年的高点（约 20）持续下滑，至 2025 年已降至 6-8 的区间，领先优势大幅收窄。
- **Kredivo**：趋势相对平稳，与下滑后的 Akulaku 在 2025 年形成高度接近的态势（兴趣值 6-8）。
- **Shopee Paylater**：下降趋势最为明显，从 2021 年与 Kredivo 持平（8-9），一路下滑至 2025 年的 2-3 区间，已明显脱离第一梯队。

当前态势：截至 2025 年，**Akulaku** **Kredivo** 的搜索兴趣值在 6-8 的区间内高度接近，呈现激烈的拉锯状态，共同占据了用户心智的主导地位。而 **Shopee Paylater** 则逐渐边缘化。

### 3.2.3 竞争格局对 YUP 的启示

这一竞争格局的分析，为理解 YUP 的入局策略提供了重要背景：

市场并非铁板一块：头部玩家（Akulaku）统治力的削弱和巨头旗下产品（Shopee Paylater）的掉队，说明这个“成熟赛道”内部的用户忠诚度可能并不稳固，市场格局仍在动态变化中。这为新进入者留下了机会窗口。

## 3.3 寻找 YUP 的潜在用户市场空白

YUP (Card) vs Akulaku vs kartu kredit(传统信用卡)

差异化定位：用户是将 YUP 视作另一个“Akulaku”，还是视作对“传统信用卡”的替代？YUP 目前的搜索量在竞品中处于什么位置？

### 3.3.1 画图

```
创建图表：寻找 YUP 的潜在空白
fig2 = go.Figure()

添加线条：寻找 YUP 的潜在空白
对比 YUP, Akulaku, 传统信用卡(kartu kredit)
fig2.add_trace(
 go.Scatter(
 x=df2['date'],
```

```

 y=df2['yup_interest'],
 mode='lines',
 name='YUP',
 line=dict(color='blue', dash='dash'), # 用虚线突出 YUP
 hovertemplate='日期: %{x}
兴趣值: %{y}<extra></extra>'
)
)
fig2.add_trace(
 go.Scatter(
 x=df2['date'],
 y=df2['akulaku_interest'],
 mode='lines',
 name='Akulaku',
 line=dict(color='red'),
 hovertemplate='日期: %{x}
兴趣值: %{y}<extra></extra>'
)
)
fig2.add_trace(
 go.Scatter(
 x=df2['date'],
 y=df2['credit_card_interest'],
 mode='lines',
 name='credit card',
 line=dict(color='grey'),
 hovertemplate='日期: %{x}
兴趣值: %{y}<extra></extra>'
)
)
更新图表的布局
fig2.update_layout(
 title='印尼数字信贷市场：YUP 与传统信用卡及头部玩家对比 (2021-2025)' ,
 hovermode='x unified',
 height=500,
 width=600,
 xaxis_title="日期",
 yaxis_title="相对搜索兴趣值"
)
fig2.show()

```

### 3.3.2 分析 YUP 的差异化定位：YUP vs Akulaku vs 传统信用卡

为了弄清 YUP 在用户心中的位置，我将其搜索兴趣与先用后付巨头产品（Akulaku）及传统信用卡进行了直接对比。

数据对比鲜明：

- 传统信用卡（Kartu Kredit）：需求基本盘非常稳固，过去五年搜索兴趣值始终在高位（60-80 区间）平稳波动，是绝对的市场关注中心。
- Akulaku：作为 BNPL 代表，其搜索兴趣从 2021 年的 20 持续下滑至 2025 年的 8，与传统信用卡的关注度完全不在一个量级。

- **YUP**：其搜索兴趣值在过去五年间始终为 0，与另外两者形成巨大断层。

核心结论：数据给出了非常清晰的答案：用户既没有将 YUP 视为另一个“Akulaku”，也没有将其当作“传统信用卡”的替代品。在主动搜索行为构成的大众认知层面，YUP 的声量几乎不存在，处于一个绝对的空白市场。

### 3.3.3 Yup 对标的市场空白

这个“空白”恰恰强力印证了我们之前的多个发现：

1. 它解释了增长模式：YUP 的业务增长（月发卡近 10 万张）几乎不依赖用户的主动线上搜索（声量为 0），必定依赖于其他渠道（如线下合作、场景嵌入）。
2. 它定义了市场机会：这个空白，可能正是那部分“有稳定收入、不满足传统信用卡条件、又不会主动搜索 BNPL 产品”的工薪阶层。YUP 用“虚拟信用卡”这个独特产品，尝试定义并服务这个未被线上声量覆盖的“沉默市场”。

因此，YUP 并非“挑战巨头”，而更像“发现并定义新需求”。它的挑战在于如何教育这个空白市场，而它的机遇也在于此——如果成功，它将拥有一个自己定义的、竞争相对缓和的细分领域。

## 3.4 洞察细分需求

### 3.4.1 画图

`pinjaman tanpa kartu kredit (无信用卡贷款) vs pinjaman cepat online (快速在线贷款)`

---

需求痛点调查：用户是否在搜索更具体、更应急、门槛更低的贷款产品？

```
强制将这两列转换为统一的数值类型（浮点数），非数字会变为NaN
df['loan_without_credit_card_interest'] = pd.to_numeric(
 df['loan_without_credit_card_interest'], errors='coerce')
df['fast_online_loan_interest'] =
pd.to_numeric(df['fast_online_loan_interest'],
errors='coerce')

处理转换后可能产生的缺失值（用前一个有效值填充）
df[['loan_without_credit_card_interest', 'fast_online_loan_interest']]
\
= df[['loan_without_credit_card_interest',
'fast_online_loan_interest']].ffill()

绘图代码
fig2 = px.line(df,
 x='date',
 y=['loan_without_credit_card_interest', \
 'fast_online_loan_interest'],
 title='细分：印尼市场对“无信用卡贷款”与“在线快速贷款”的搜索兴趣对比
(过去5年)',
```

```

 labels={'value': '相对搜索兴趣值', 'date': '日期',
'variable': '搜索关键词'},
 color_discrete_map={
 'loan_without_credit_card_interest': 'green',
 'fast_online_loan_interest': 'orange'
 })
fig2.update_layout(hovermode='x unified')

在显示前调整图表尺寸，使其适合 Colab 打印展示
fig2.update_layout(
 height=400, # 设置合适的高度
 width=800, # 设置合适的宽度
 margin=dict(l=50, r=50, t=80, b=80), # 调整边距
 title_font=dict(size=16), # 调整标题大小
 font=dict(size=11), # 调整整体字体大小
 xaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10),
),
 yaxis=dict(
 title_font=dict(size=12),
 tickfont=dict(size=10)
),
 legend=dict(
 title_font=dict(size=11),
 font=dict(size=10),
 yanchor="top",
 y=0.99,
 xanchor="left",
 x=0.01,
 bgcolor='rgba(255, 255, 255, 0.8)',
 title_text="搜索关键词" # 添加图例标题
)
)

调整线条样式，使其更清晰
fig2.update_traces(
 line=dict(width=2),
 marker=dict(size=4)
)
fig2.show()

```

### 3.4.2 细分需求分析：搜索行为揭示的市场“静默区”

通过对 `pinjaman cepat online`(在线快速贷款) 和 `pinjaman tanpa kartu kredit`(无信用卡贷款) 的搜索趋势分析，我们发现了一个有趣的现象：

- 应急性需求降温：代表应急、快速获取资金的 `pinjaman cepat online` 搜索兴趣在 2021 年达到短期高点后持续回落，并长期维持在低位（兴趣值 1-2）。这表明用户对这类可能伴随高成本或风险的短期贷款产品的热度正在减退。
- 具体性需求几乎不存在：而更具体指向“无信用卡贷款”的搜索词，其兴趣值在过去五年间几乎全程为 0。这强烈暗示，在印尼普通网民中，并不存在一个主动、大规模地搜索“没有信用卡如何贷款”的线上行为。

### 3.4.3 核心洞察与项目连接

这为我们理解 YUP 的定位提供了关键线索：

1. 市场存在“认知空白”：大众没有主动搜索“无信用卡贷款”，很可能意味着他们尚未普遍意识到“没有信用卡也能获得可靠信贷服务”是一种可能。他们的选择可能被局限在“申请传统信用卡（被拒）”或“寻找高成本的快速贷款”之间。
2. YUP 在创造需求，而非满足显性需求：YUP 的商业模式，很可能不是去争夺那些已经在搜索“快速贷款”的、对价格和风险更敏感的用户。它在教育并满足一个“未被搜索”的潜在需求。

结论：数据表明，印尼市场在“传统信用卡”和“高风险快速贷款”之间，存在一个巨大的“沉默的中间市场”。存在一个很大的工薪阶层群体，他们需要信贷服务，但既不满足传统信用卡的条件，又不会（或不敢）去搜那些高风险的快速贷款。他们是一个“沉默的”潜在市场。YUP 这类产品，可能正是通过线下合作（比如和便利店、快餐店一起推）的方式，绕开了线上激烈的流量竞争，直接找到了这群“沉默”的用户，并用“虚拟信用卡”这种他们更容易接受的产品形式，满足了他们的需求。

## 四：SWOT 模型分析

### 4.1 YUP 产品的 SWOT 详细描述

#### 内部优势（Strength）

1. 产品：虚拟+实体 Visa 卡双模式
2. 风控：Finture Tech 公司自研神经网络模型，坏账率低
3. 素材：现有用户来自线下快餐店合作，广告中食物素材获客成本低

#### 内部劣势（Weakness）

1. 品牌：搜索量为零，无知名度
2. 教育：需要教用户接受“新品类”
3. 业务：只做信贷，依赖印尼单一市场

#### 外部机会（Opportunities）

1. 市场：2-3 千万工薪中低收入阶层缺正规信贷服务
2. 竞争：BNPL（先用后付）玩家增长乏力，用户在找替代产品
3. 环境：数字支付习惯已养成

## 外部威胁 ( Threats )

- 巨头 : Gojek、Shopee 可能进场
- 政策 : 印尼监管和经济变化风险 , 效果广告素材容易被渠道禁
- 模仿 : 市场验证后会涌入大量竞争者

## 4.2 SWOT 画图

```
from IPython.display import display, HTML

swot_html = """
<style>
.swot-table {
 width: 100%;
 border-collapse: collapse;
 margin: 20px 0;
 font-family: Arial, sans-serif;
}
.swot-table th, .swot-table td {
 border: 2px solid #ddd;
 padding: 15px;
 text-align: left;
 vertical-align: top;
}
.swot-table th {
 background-color: #f8f9fa;
 font-weight: bold;
 text-align: center;
}
.strengths { background-color: #d4edda; border-left: 5px solid #28a745; }
.weaknesses { background-color: #fff3cd; border-left: 5px solid #ffc107; }
.opportunities { background-color: #d1ecf1; border-left: 5px solid #17a2b8; }
.threats { background-color: #f8d7da; border-left: 5px solid #dc3545; }
.badge {
 display: inline-block;
 padding: 3px 8px;
 border-radius: 12px;
 font-size: 12px;
 font-weight: bold;
 margin-right: 5px;
}
.data-badge { background-color: #6c757d; color: white; }
</style>

<h3>[] Yup 信用卡 SWOT 分析表</h3>
<table class="swot-table">
```

```
<tr>
 <th width="50%">内部因素</th>
 <th width="50%">外部因素</th>
</tr>
<tr>
 <td>
 <div class="strengths">
 <h4>优势 (Strengths)</h4>
 <p>数据支撑精准空白市场定位</p>
 <p>广告数据高效获客渠道 (CPA $7-12)</p>
 <p>趋势分析 </p>
 <p>素材测试食物素材转化率高</p>
 </div>
 </td>
 <td>
 <div class="opportunities">
 <h4>机会 (Opportunities)</h4>
 <p>宏观数据3000 万市场缺口</p>
 <p>竞品分析竞争格局有利</p>
 <p>趋势分析数字基础成熟</p>
 <p>宏观数据互联网 > 金融普及</p>
 </div>
 </td>
</tr>
<tr>
 <td>
 <div class="weaknesses">
 <h4>劣势 (Weaknesses)</h4>
 <p>搜索数据品牌认知为零</p>
 <p>渠道数据渠道单一风险</p>
 <p>产品分析产品形态单一</p>
 <p>广告数据Meta/TikTok 渠道无效</p>
 </div>
 </td>
 <td>
 <div class="threats">
 <h4>威胁 (Threats)</h4>
 <p>竞争分析竞品可能跟进</p>
 </div>
 </td>
</tr>
```

```

</p>
 <p>行业研究监管政策风险
</p>
 <p>经济数据经济波动风险
</p>
 <p>渠道分析渠道成本上升
风险</p>
</div>
</td>
</tr>
</table>

<p><small>□ 分析说明 : 本 SWOT 分析基于世界银行宏观数
据、Google Trends 搜索数据、广告投放数据综合得出，每个观点都有具体数据支
撑。</small></p>
"""

display(HTML(swot_html))

保存为HTML文件
with open('yup_swot_analysis.html', 'w', encoding='utf-8') as f:
 f.write(swot_html)
print("□ SWOT 分析 HTML 文件已保存为 'yup_swot_analysis.html'")

<IPython.core.display.HTML object>
□ SWOT 分析 HTML 文件已保存为 'yup_swot_analysis.html'

```

## 五：（对后续执行）三大策略建议

### 5.1 渠道策略

#### 1. 继续大力投入 Google ()

- 为什么：因为 Google 搜索广告 CPA 只有\$7-12，是行业 1/3
- 怎么做：把 Google 预算从现在的水平提升到总预算的 60%
- 具体行动：
  - a. 测试"生活化"素材（不只是炸鸡，还有外卖、打车、网购场景）
  - b. 买"信用卡申请"、"在线贷款"这些关键词
  - c. 记录哪些素材转化最好，大量复制

#### 2. 改变 Meta/TikTok 的用法（现在效果不好）

- 为什么：这两个渠道转化率低，但用户多
- 怎么做：不做直接转化广告，改做"品牌教育"
- 具体行动：
  - a. 拍短视频教用户"如何建立信用记录"

- b. 分享真实用户故事（比如：被银行拒绝 3 次，终于有了 Yup 卡）
- c. 不要直接卖卡，先让用户了解产品价值

## 5.2 产品策略

### 1. 推出"额度成长计划"（解决用户额度低的痛点）

- 怎么做：用户用卡 6 个月还款良好 → 自动提额 20%
- 好处：用户有动力好好用卡，不会因为额度低就放弃

### 2. 增加使用场景（让卡更有用）

- 怎么做：
  - 和便利店(Indomaret)合作：用 Yup 卡付款有折扣
  - 和快餐店(KFC)合作：消费满额送优惠券
  - 开发线上商城：可以用 Yup 卡分期买东西
- 好处：用户用得越多，Yup 赚得越多

### 3. ()

- 怎么做：
  - Yup Lite：500 元额度，给学生/新手
  - Yup Plus：5000 元额度，给稳定收入者
- 好处：覆盖不同需求的用户

## 5.3 市场策略

### 1. 解决"品牌没人知道"的问题

- 现状：Google Trends 显示 Yup 搜索量为 0
- 目标：1年内让搜索量达到 1000+
- 怎么做：
  - 在社交媒体讲"被银行拒绝者"的故事
  - 合作本地网红展示真实使用场景 - 本地化 KOL 运营
  - 做线下活动（在便利店设申请点）

### 2. 确定市场定位

- 不要说："我们是另一个贷款 App"
- 要说："你的第一张正规信用卡"
- 重点宣传："用 Yup 建立信用记录"

### 3. 考虑未来扩张

- 现在：专注印尼（市场足够大）
- 1年后：考虑菲律宾（情况类似）
- 3年后：覆盖 2-3 个东南亚国家

# 六：对 Yup 的战略建议 (SWOT 框架)

## 战略建议表 (SO+WO+ST)

策略维度	核心问题	关键建议	预期效果
SO 策略快速测试与规模化	Google 广告素材频繁被封	1. 建立“素材矩阵”分散风险 2. 小预算快速测试迭代 3. 合规优化前置：了解谷歌政策 & 建立备用素材库	• 稳定获客渠道·降低封禁风险·CPA 稳定在 \$15 以下
WO 策略从流量到用户资产	过度依赖效果广告，品牌资产为零	1. 社媒运营：广告引流→私域沉淀 2. 建立用户信任资产 3. 产品新功能、活动开发	• 降低渠道依赖·提升品牌认知·提高用户 LTV
ST 策略建立运营护城河	易被模仿，竞争壁垒低	1. 开发素材效果预测模型 2. 与本地 KOL/商户建立深度合作 3. 建立合规审核 checklist	• 提高模仿门槛·建立本地优势·形成可复制模式

# 七：项目的局限性与后续研究建议

## 项目数据来源和研究方法限制

限制类别	具体说明	影响
数据来源限制	主要依赖世界银行、Google Trends 等公开二手数据，缺乏公司内部数据与一手问卷调查数据	结论的时效性与针对性可能不足，无法反映最新市场动态
商业数据缺失	出于保密考虑，未使用 Yup 实际广告投放的详细效果数据（如分渠道 CPA、转化漏斗）	预算分配与渠道建议更多基于逻辑推演，而非精准数据测算
研究方法局限	未包含对目标用户（“被银行拒绝的工薪阶层”）的深度访谈或焦点小组讨论	对用户真实动机缺乏第一视角洞察
竞品分析深度	侧重于宏观市场格局与搜索趋势，未对竞品（如其他 BNPL 产品）进行具体产品与案例拆解	可能遗漏竞品的关键成功因素或差异化机会

注：上述限制不影响核心市场机会（“沉默市场”）的判断，但建议在后续执行中通过补充研究予以验证和细化。

## 后续可开展的项目拓展

1. App Store 评价分析：系统抓取并分析 Yup 及主要竞品的用户评价（最近 6 个月），识别用户核心赞扬点、主要投诉、需求痛点

2. 轻量级问卷调查：询问用户知道 Yup 的途径，申请的原因和顾虑，了解核心获客渠道和建立一手数据库
3. 用户深度访谈：对 10-15 名典型用户（新用户、活跃用户、流失用户）进行焦点采访，建立细化用户画像和用户转化/流失地图

## 八：项目结论

### 总结与核心洞察

#### 1. 分析起源

在代理 Yup 信用卡的广告投放时，我发现了一个反常现象：

- Meta/TikTok 转化效果很差
- 但同类图片素材在 Google 渠道冷启动期间获客成本极低（周平均 CPA = 行业水准的 1/3）

#### 2. 推测产品性质与受众

结合客户历史爆款视频素材风格简陋（不同于金融产品的专业度），以及 Yup 的广告语 **For those who've been told NO we say YUP**，我推测它服务的可能是“被传统银行拒绝”的特殊人群。

#### 3. 验证过程

为验证这个猜想，我从两个维度进行了分析：

- 宏观市场：使用世界银行数据分析印尼的金融包容性、互联网普及率、信贷环境、人群收入结构
- 用户行为：使用 Google Trends 分析信贷产品的搜索趋势和竞争格局

#### 4. 核心发现

Yup 服务的是一个“沉默市场”——约 3000 万有中等教育背景、有稳定收入，但被传统银行拒绝的印尼工薪阶层。

#### 5. 关键证据

- 印尼 31.7% 中等教育人群无正规金融账户
- Yup 品牌搜索量几乎为 0，用户不主动寻找此类产品
- 传统信用卡搜索量大，但 BNPL 产品关注度下降
- 印尼私人信贷/GDP 仅 38.7%，远低于邻国

#### 6. 战略建议

- 产品定位：不要将自己定位为“另一个贷款 App”，而是“你的第一张正经信用卡”
- 市场沟通：
  - 主打“品类教育”：教用户什么是信用身份，如何建立信用记录

- 避免竞品对比：不提 BNPL 或快速贷款，定义新品类
  - 素材方向：继续使用“生活化”素材（如食物），降低金融决策的心理防线
  - 预算分配：
    - 60%：Google 渠道投入（验证有效的渠道）
    - 15%：Meta/TikTok 转为品牌教育，而非直接转化
    - 25%：深化线下合作（便利店、快餐店等场景获客）
- （注：此预算比例为策略性示范，实际商业预算需根据广告效果、KOL 运营成本及用户 LTV 等核心数据进行精确核算。）

## 7. 商业价值

这个分析验证了 Yup 的空白市场定位是正确的，并给出了具体可执行的优化方向。对于金融科技公司来说，找准“沉默市场”并有效触达，是避开红海竞争、实现快速增长的关键策略。

---

分析基于世界银行公开数据与 Google Trends 搜索数据，结论仅供参考。项目完整代码与分析过程见本 Colab 文件。