<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>抖音带货数据分析系统</title>

<script src="https://cdn.tailwindcss.com"></script>

<link href="https://cdn.jsdelivr.net/npm/font-awesome@4.7.0/css/font-awesome.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/chart.js@4.4.8/dist/chart.umd.min.js"></script>

<script>

tailwind.config = {

theme: {

extend: {

colors: {

primary: '#FE2C55', // 抖音红

secondary: '#222222', // 抖音黑

accent: '#00F2EA', // 抖音青

neutral: '#F5F5F5', // 背景灰

},

fontFamily: {

sans: ['Inter', 'system-ui', 'sans-serif'],

},

}

}

}

</script>

<style type="text/tailwindcss">

@layer utilities {

.content-auto {

content-visibility: auto;

}

.card-shadow {

box-shadow: 0 4px 20px rgba(0, 0, 0, 0.08);

}

.chart-container {

position: relative;

height: 300px;

}

}

</style>

</head>

<body class="bg-neutral text-secondary min-h-screen">

<!-- 顶部导航 -->

<header class="bg-white card-shadow fixed w-full top-0 z-50">

<div class="container mx-auto px-4 py-3 flex justify-between items-center">

<div class="flex items-center space-x-2">

<i class="fa fa-line-chart text-primary text-2xl"></i>

<h1 class="text-xl font-bold">抖音带货数据分析</h1>

</div>

<nav class="hidden md:flex space-x-6">

<a href="#dashboard" class="font-medium hover:text-primary transition-colors">数据概览</a>

<a href="#sales" class="font-medium hover:text-primary transition-colors">销售分析</a>

<a href="#audience" class="font-medium hover:text-primary transition-colors">观众互动</a>

<a href="#products" class="font-medium hover:text-primary transition-colors">商品表现</a>

<a href="#code" class="font-medium hover:text-primary transition-colors">代码实现</a>

</nav>

<button class="md:hidden text-secondary text-xl">

<i class="fa fa-bars"></i>

</button>

</div>

</header>

<!-- 主内容区 -->

<main class="container mx-auto px-4 pt-24 pb-12">

<!-- 数据概览 -->

<section id="dashboard" class="mb-12">

<h2 class="text-[clamp(1.5rem,3vw,2.5rem)] font-bold mb-6 border-b-2 border-primary pb-2">数据概览</h2>

<!-- 关键指标卡片 -->

<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-6 mb-8">

<div class="bg-white rounded-xl p-6 card-shadow transform hover:scale-[1.02] transition-transform">

<div class="flex justify-between items-start mb-4">

<div>

<p class="text-gray-500 text-sm">总销售额</p>

<h3 class="text-3xl font-bold">¥258,643</h3>

</div>

<div class="p-3 rounded-full bg-primary/10">

<i class="fa fa-shopping-cart text-primary"></i>

</div>

</div>

<div class="flex items-center text-sm">

<span class="text-green-500 flex items-center">

<i class="fa fa-arrow-up mr-1"></i> 12.5%

</span>

<span class="text-gray-500 ml-2">vs 上周</span>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow transform hover:scale-[1.02] transition-transform">

<div class="flex justify-between items-start mb-4">

<div>

<p class="text-gray-500 text-sm">观众人数</p>

<h3 class="text-3xl font-bold">124,857</h3>

</div>

<div class="p-3 rounded-full bg-accent/10">

<i class="fa fa-users text-accent"></i>

</div>

</div>

<div class="flex items-center text-sm">

<span class="text-green-500 flex items-center">

<i class="fa fa-arrow-up mr-1"></i> 8.3%

</span>

<span class="text-gray-500 ml-2">vs 上周</span>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow transform hover:scale-[1.02] transition-transform">

<div class="flex justify-between items-start mb-4">

<div>

<p class="text-gray-500 text-sm">转化率</p>

<h3 class="text-3xl font-bold">4.2%</h3>

</div>

<div class="p-3 rounded-full bg-blue-100">

<i class="fa fa-exchange text-blue-500"></i>

</div>

</div>

<div class="flex items-center text-sm">

<span class="text-red-500 flex items-center">

<i class="fa fa-arrow-down mr-1"></i> 1.2%

</span>

<span class="text-gray-500 ml-2">vs 上周</span>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow transform hover:scale-[1.02] transition-transform">

<div class="flex justify-between items-start mb-4">

<div>

<p class="text-gray-500 text-sm">客单价</p>

<h3 class="text-3xl font-bold">¥207</h3>

</div>

<div class="p-3 rounded-full bg-purple-100">

<i class="fa fa-rmb text-purple-500"></i>

</div>

</div>

<div class="flex items-center text-sm">

<span class="text-green-500 flex items-center">

<i class="fa fa-arrow-up mr-1"></i> 5.7%

</span>

<span class="text-gray-500 ml-2">vs 上周</span>

</div>

</div>

</div>

<!-- 趋势图表 -->

<div class="grid grid-cols-1 lg:grid-cols-3 gap-6">

<div class="bg-white rounded-xl p-6 card-shadow lg:col-span-2">

<h3 class="text-lg font-semibold mb-4">销售额趋势</h3>

<div class="chart-container">

<canvas id="salesChart"></canvas>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">销售渠道分布</h3>

<div class="chart-container">

<canvas id="channelChart"></canvas>

</div>

</div>

</div>

</section>

<!-- 销售分析 -->

<section id="sales" class="mb-12">

<h2 class="text-[clamp(1.5rem,3vw,2.5rem)] font-bold mb-6 border-b-2 border-primary pb-2">销售分析</h2>

<div class="grid grid-cols-1 lg:grid-cols-2 gap-6">

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">按小时销售分布</h3>

<div class="chart-container">

<canvas id="hourlySalesChart"></canvas>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">地域销售TOP10</h3>

<div class="chart-container">

<canvas id="regionChart"></canvas>

</div>

</div>

</div>

<!-- 销售明细表格 -->

<div class="bg-white rounded-xl p-6 card-shadow mt-6">

<div class="flex justify-between items-center mb-4">

<h3 class="text-lg font-semibold">销售订单明细</h3>

<div class="relative">

<input type="text" placeholder="搜索订单..." class="pl-10 pr-4 py-2 rounded-lg border border-gray-200 focus:outline-none focus:ring-2 focus:ring-primary/20">

<i class="fa fa-search absolute left-3 top-1/2 transform -translate-y-1/2 text-gray-400"></i>

</div>

</div>

<div class="overflow-x-auto">

<table class="w-full">

<thead>

<tr class="border-b border-gray-200">

<th class="py-3 px-4 text-left font-medium text-gray-500">订单ID</th>

<th class="py-3 px-4 text-left font-medium text-gray-500">用户ID</th>

<th class="py-3 px-4 text-left font-medium text-gray-500">商品名称</th>

<th class="py-3 px-4 text-left font-medium text-gray-500">金额</th>

<th class="py-3 px-4 text-left font-medium text-gray-500">时间</th>

<th class="py-3 px-4 text-left font-medium text-gray-500">状态</th>

</tr>

</thead>

<tbody>

<tr class="border-b border-gray-100 hover:bg-gray-50 transition-colors">

<td class="py-3 px-4">#ORD-20230512-001</td>

<td class="py-3 px-4">USER-8765</td>

<td class="py-3 px-4">抖音爆款口红套装</td>

<td class="py-3 px-4">¥199.00</td>

<td class="py-3 px-4">2023-05-12 14:32</td>

<td class="py-3 px-4"><span class="px-2 py-1 bg-green-100 text-green-800 rounded-full text-xs">已完成</span></td>

</tr>

<tr class="border-b border-gray-100 hover:bg-gray-50 transition-colors">

<td class="py-3 px-4">#ORD-20230512-002</td>

<td class="py-3 px-4">USER-9876</td>

<td class="py-3 px-4">无线蓝牙耳机</td>

<td class="py-3 px-4">¥299.00</td>

<td class="py-3 px-4">2023-05-12 15:15</td>

<td class="py-3 px-4"><span class="px-2 py-1 bg-green-100 text-green-800 rounded-full text-xs">已完成</span></td>

</tr>

<tr class="border-b border-gray-100 hover:bg-gray-50 transition-colors">

<td class="py-3 px-4">#ORD-20230512-003</td>

<td class="py-3 px-4">USER-6543</td>

<td class="py-3 px-4">智能手表</td>

<td class="py-3 px-4">¥599.00</td>

<td class="py-3 px-4">2023-05-12 16:08</td>

<td class="py-3 px-4"><span class="px-2 py-1 bg-yellow-100 text-yellow-800 rounded-full text-xs">配送中</span></td>

</tr>

<tr class="border-b border-gray-100 hover:bg-gray-50 transition-colors">

<td class="py-3 px-4">#ORD-20230512-004</td>

<td class="py-3 px-4">USER-7654</td>

<td class="py-3 px-4">网红美食大礼包</td>

<td class="py-3 px-4">¥159.00</td>

<td class="py-3 px-4">2023-05-12 16:45</td>

<td class="py-3 px-4"><span class="px-2 py-1 bg-red-100 text-red-800 rounded-full text-xs">已退款</span></td>

</tr>

<tr class="hover:bg-gray-50 transition-colors">

<td class="py-3 px-4">#ORD-20230512-005</td>

<td class="py-3 px-4">USER-8765</td>

<td class="py-3 px-4">抖音爆款T恤</td>

<td class="py-3 px-4">¥89.00</td>

<td class="py-3 px-4">2023-05-12 17:20</td>

<td class="py-3 px-4"><span class="px-2 py-1 bg-green-100 text-green-800 rounded-full text-xs">已完成</span></td>

</tr>

</tbody>

</table>

</div>

<div class="flex justify-between items-center mt-4">

<p class="text-sm text-gray-500">显示 1-5 条，共 124 条</p>

<div class="flex space-x-2">

<button class="px-3 py-1 rounded-lg border border-gray-200 text-gray-500 hover:bg-gray-100 transition-colors">上一页</button>

<button class="px-3 py-1 rounded-lg bg-primary text-white hover:bg-primary/90 transition-colors">1</button>

<button class="px-3 py-1 rounded-lg border border-gray-200 text-gray-500 hover:bg-gray-100 transition-colors">2</button>

<button class="px-3 py-1 rounded-lg border border-gray-200 text-gray-500 hover:bg-gray-100 transition-colors">3</button>

<button class="px-3 py-1 rounded-lg border border-gray-200 text-gray-500 hover:bg-gray-100 transition-colors">下一页</button>

</div>

</div>

</div>

</section>

<!-- 观众互动 -->

<section id="audience" class="mb-12">

<h2 class="text-[clamp(1.5rem,3vw,2.5rem)] font-bold mb-6 border-b-2 border-primary pb-2">观众互动分析</h2>

<div class="grid grid-cols-1 lg:grid-cols-3 gap-6">

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">观众性别分布</h3>

<div class="chart-container">

<canvas id="genderChart"></canvas>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">观众年龄分布</h3>

<div class="chart-container">

<canvas id="ageChart"></canvas>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">互动行为分布</h3>

<div class="chart-container">

<canvas id="interactionChart"></canvas>

</div>

</div>

</div>

<!-- 互动趋势 -->

<div class="bg-white rounded-xl p-6 card-shadow mt-6">

<h3 class="text-lg font-semibold mb-4">直播时段互动趋势</h3>

<div class="chart-container">

<canvas id="interactionTrendChart"></canvas>

</div>

</div>

</section>

<!-- 商品表现 -->

<section id="products" class="mb-12">

<h2 class="text-[clamp(1.5rem,3vw,2.5rem)] font-bold mb-6 border-b-2 border-primary pb-2">商品表现分析</h2>

<div class="grid grid-cols-1 lg:grid-cols-2 gap-6">

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">商品销售额TOP10</h3>

<div class="chart-container">

<canvas id="productSalesChart"></canvas>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">商品转化率对比</h3>

<div class="chart-container">

<canvas id="conversionChart"></canvas>

</div>

</div>

</div>

<!-- 商品分类分析 -->

<div class="bg-white rounded-xl p-6 card-shadow mt-6">

<h3 class="text-lg font-semibold mb-4">商品类别销售分析</h3>

<div class="chart-container">

<canvas id="categoryChart"></canvas>

</div>

</div>

</section>

<!-- 代码实现 -->

<section id="code" class="mb-12">

<h2 class="text-[clamp(1.5rem,3vw,2.5rem)] font-bold mb-6 border-b-2 border-primary pb-2">抖音带货数据分析代码实现</h2>

<div class="bg-white rounded-xl p-6 card-shadow">

<div class="flex justify-between items-center mb-4">

<h3 class="text-lg font-semibold">Python数据分析代码</h3>

<div class="flex space-x-2">

<button class="px-3 py-1 rounded-lg border border-gray-200 text-gray-500 hover:bg-gray-100 transition-colors">

<i class="fa fa-copy mr-1"></i> 复制

</button>

<button class="px-3 py-1 rounded-lg bg-primary text-white hover:bg-primary/90 transition-colors">

<i class="fa fa-download mr-1"></i> 下载

</button>

</div>

</div>

<div class="bg-gray-900 text-gray-100 rounded-lg p-4 overflow-x-auto">

<pre><code class="language-python">import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from datetime import datetime, timedelta

import requests

import json

import time

from io import StringIO

from sklearn.ensemble import RandomForestRegressor

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

from sklearn.metrics import mean\_squared\_error, r2\_score

# 设置中文字体显示

plt.rcParams['font.sans-serif'] = ['SimHei']

plt.rcParams['axes.unicode\_minus'] = False

# API密钥配置

API\_KEY = "your\_douyin\_api\_key"

SECRET = "your\_douyin\_api\_secret"

class DouYinEcommerceAnalyzer:

def \_\_init\_\_(self, api\_key, secret):

"""初始化抖音电商分析器"""

self.api\_key = api\_key

self.secret = secret

self.access\_token = None

self.token\_expires\_at = None

def get\_access\_token(self):

"""获取抖音API访问令牌"""

if self.access\_token and datetime.now() < self.token\_expires\_at:

return self.access\_token

# 实际项目中，这里应该调用抖音开放平台的认证API

# 此处仅为示例

self.access\_token = "sample\_access\_token"

self.token\_expires\_at = datetime.now() + timedelta(hours=2)

return self.access\_token

def fetch\_live\_stream\_data(self, start\_date, end\_date, page=1, page\_size=100):

"""获取直播数据"""

token = self.get\_access\_token()

# 模拟API请求

# 实际项目中，这里应该调用抖音开放平台的直播数据API

time.sleep(1) # 模拟网络延迟

# 生成模拟数据

dates = pd.date\_range(start=start\_date, end=end\_date)

data = []

for date in dates:

for hour in range(24):

viewers = np.random.randint(100, 5000)

sales = np.random.randint(0, int(viewers \* 0.05))

revenue = sales \* np.random.randint(50, 300)

likes = np.random.randint(0, viewers \* 10)

comments = np.random.randint(0, viewers)

shares = np.random.randint(0, int(viewers \* 0.05))

data.append({

'date': date.strftime('%Y-%m-%d'),

'hour': hour,

'viewers': viewers,

'sales': sales,

'revenue': revenue,

'likes': likes,

'comments': comments,

'shares': shares

})

return pd.DataFrame(data)

def fetch\_product\_data(self, start\_date, end\_date):

"""获取商品数据"""

token = self.get\_access\_token()

# 模拟API请求

time.sleep(1)

# 生成模拟数据

products = [

{"id": 1001, "name": "抖音爆款口红套装", "category": "美妆个护", "price": 199, "cost": 80},

{"id": 1002, "name": "无线蓝牙耳机", "category": "数码家电", "price": 299, "cost": 120},

{"id": 1003, "name": "智能手表", "category": "数码家电", "price": 599, "cost": 250},

{"id": 1004, "name": "网红美食大礼包", "category": "食品饮料", "price": 159, "cost": 70},

{"id": 1005, "name": "抖音爆款T恤", "category": "服装配饰", "price": 89, "cost": 35},

{"id": 1006, "name": "家用健身器材", "category": "运动户外", "price": 399, "cost": 180},

{"id": 1007, "name": "多功能料理锅", "category": "家居日用", "price": 299, "cost": 130},

{"id": 1008, "name": "时尚太阳镜", "category": "服装配饰", "price": 129, "cost": 50},

{"id": 1009, "name": "高端护肤品套装", "category": "美妆个护", "price": 499, "cost": 200},

{"id": 1010, "name": "智能扫地机器人", "category": "数码家电", "price": 1299, "cost": 500}

]

dates = pd.date\_range(start=start\_date, end=end\_date)

data = []

for date in dates:

for product in products:

sales = np.random.randint(0, 100)

impressions = np.random.randint(sales \* 10, sales \* 100)

data.append({

'date': date.strftime('%Y-%m-%d'),

'product\_id': product['id'],

'product\_name': product['name'],

'category': product['category'],

'price': product['price'],

'cost': product['cost'],

'sales': sales,

'revenue': sales \* product['price'],

'impressions': impressions,

'conversion\_rate': sales / impressions if impressions > 0 else 0

})

return pd.DataFrame(data)

def fetch\_user\_data(self, start\_date, end\_date):

"""获取用户数据"""

token = self.get\_access\_token()

# 模拟API请求

time.sleep(1)

# 生成模拟数据

dates = pd.date\_range(start=start\_date, end=end\_date)

data = []

for date in dates:

for hour in range(24):

total\_users = np.random.randint(100, 5000)

# 性别分布

male\_ratio = np.random.uniform(0.3, 0.7)

male\_users = int(total\_users \* male\_ratio)

female\_users = total\_users - male\_users

# 年龄分布

age\_groups = {

'18岁以下': int(total\_users \* 0.1),

'18-24岁': int(total\_users \* 0.3),

'25-34岁': int(total\_users \* 0.35),

'35-44岁': int(total\_users \* 0.15),

'45岁以上': int(total\_users \* 0.1)

}

# 地区分布

regions = {

'广东': int(total\_users \* 0.15),

'江苏': int(total\_users \* 0.12),

'浙江': int(total\_users \* 0.12),

'山东': int(total\_users \* 0.09),

'河南': int(total\_users \* 0.08),

'四川': int(total\_users \* 0.08),

'湖北': int(total\_users \* 0.07),

'福建': int(total\_users \* 0.06),

'湖南': int(total\_users \* 0.06),

'其他': total\_users - sum(regions.values())

}

data.append({

'date': date.strftime('%Y-%m-%d'),

'hour': hour,

'total\_users': total\_users,

'male\_users': male\_users,

'female\_users': female\_users,

'age\_groups': json.dumps(age\_groups),

'regions': json.dumps(regions)

})

return pd.DataFrame(data)

def preprocess\_data(self, live\_data, product\_data, user\_data):

"""数据预处理"""

# 处理日期时间

live\_data['datetime'] = pd.to\_datetime(live\_data['date'] + ' ' + live\_data['hour'].astype(str) + ':00:00')

live\_data['day\_of\_week'] = live\_data['datetime'].dt.dayofweek

live\_data['is\_weekend'] = live\_data['day\_of\_week'].apply(lambda x: 1 if x >= 5 else 0)

product\_data['date'] = pd.to\_datetime(product\_data['date'])

# 处理用户数据中的JSON字段

user\_data['age\_groups'] = user\_data['age\_groups'].apply(json.loads)

user\_data['regions'] = user\_data['regions'].apply(json.loads)

# 计算直播数据的汇总指标

daily\_live\_summary = live\_data.groupby('date').agg({

'viewers': 'sum',

'sales': 'sum',

'revenue': 'sum',

'likes': 'sum',

'comments': 'sum',

'shares': 'sum'

}).reset\_index()

# 计算转化率

daily\_live\_summary['conversion\_rate'] = daily\_live\_summary['sales'] / daily\_live\_summary['viewers'] \* 100

return {

'live\_data': live\_data,

'product\_data': product\_data,

'user\_data': user\_data,

'daily\_live\_summary': daily\_live\_summary

}

def analyze\_sales\_trends(self, data):

"""分析销售趋势"""

live\_data = data['live\_data']

daily\_summary = data['daily\_live\_summary']

# 按小时分析销售情况

hourly\_sales = live\_data.groupby('hour').agg({

'sales': 'mean',

'revenue': 'mean',

'viewers': 'mean'

}).reset\_index()

# 按星期分析销售情况

weekday\_sales = live\_data.groupby('day\_of\_week').agg({

'sales': 'mean',

'revenue': 'mean',

'viewers': 'mean'

}).reset\_index()

weekday\_sales['day\_name'] = weekday\_sales['day\_of\_week'].map({

0: '周一', 1: '周二', 2: '周三', 3: '周四', 4: '周五', 5: '周六', 6: '周日'

})

# 计算同比环比

daily\_summary['date'] = pd.to\_datetime(daily\_summary['date'])

daily\_summary = daily\_summary.sort\_values('date')

# 计算环比

daily\_summary['prev\_day\_revenue'] = daily\_summary['revenue'].shift(1)

daily\_summary['mom\_growth'] = (daily\_summary['revenue'] - daily\_summary['prev\_day\_revenue']) / daily\_summary['prev\_day\_revenue'] \* 100

# 计算同比（假设同比为7天前）

daily\_summary['prev\_week\_revenue'] = daily\_summary['revenue'].shift(7)

daily\_summary['yoy\_growth'] = (daily\_summary['revenue'] - daily\_summary['prev\_week\_revenue']) / daily\_summary['prev\_week\_revenue'] \* 100

return {

'hourly\_sales': hourly\_sales,

'weekday\_sales': weekday\_sales,

'daily\_growth': daily\_summary[['date', 'revenue', 'mom\_growth', 'yoy\_growth']].dropna()

}

def analyze\_audience(self, data):

"""分析观众特征"""

user\_data = data['user\_data']

# 汇总所有日期的用户数据

all\_age\_data = []

all\_region\_data = []

for \_, row in user\_data.iterrows():

age\_data = row['age\_groups']

region\_data = row['regions']

# 展开年龄数据

for age\_group, count in age\_data.items():

all\_age\_data.append({

'date': row['date'],

'age\_group': age\_group,

'count': count

})

# 展开地区数据

for region, count in region\_data.items():

all\_region\_data.append({

'date': row['date'],

'region': region,

'count': count

})

# 转换为DataFrame

age\_df = pd.DataFrame(all\_age\_data)

region\_df = pd.DataFrame(all\_region\_data)

# 汇总年龄分布

age\_summary = age\_df.groupby('age\_group')['count'].sum().reset\_index()

age\_summary['percentage'] = age\_summary['count'] / age\_summary['count'].sum() \* 100

# 汇总地区分布

region\_summary = region\_df.groupby('region')['count'].sum().reset\_index()

region\_summary['percentage'] = region\_summary['count'] / region\_summary['count'].sum() \* 100

region\_summary = region\_summary.sort\_values('count', ascending=False).head(10)

# 性别分布

gender\_summary = user\_data.agg({

'male\_users': 'sum',

'female\_users': 'sum'

}).reset\_index()

gender\_summary.columns = ['gender', 'count']

gender\_summary['percentage'] = gender\_summary['count'] / gender\_summary['count'].sum() \* 100

return {

'age\_distribution': age\_summary,

'region\_distribution': region\_summary,

'gender\_distribution': gender\_summary

}

def analyze\_products(self, data):

"""分析商品表现"""

product\_data = data['product\_data']

# 按商品汇总

product\_summary = product\_data.groupby(['product\_id', 'product\_name', 'category']).agg({

'sales': 'sum',

'revenue': 'sum',

'impressions': 'sum',

'conversion\_rate': 'mean'

}).reset\_index()

# 计算利润

product\_summary['profit'] = product\_summary['revenue'] - product\_summary['sales'] \* product\_data.groupby('product\_id')['cost'].first().values

# 按销售额排序

top\_products\_by\_revenue = product\_summary.sort\_values('revenue', ascending=False).head(10)

# 按转化率排序

top\_products\_by\_conversion = product\_summary.sort\_values('conversion\_rate', ascending=False).head(10)

# 按类别汇总

category\_summary = product\_data.groupby('category').agg({

'sales': 'sum',

'revenue': 'sum',

'impressions': 'sum'

}).reset\_index()

category\_summary['conversion\_rate'] = category\_summary['sales'] / category\_summary['impressions'] \* 100

return {

'top\_products\_by\_revenue': top\_products\_by\_revenue,

'top\_products\_by\_conversion': top\_products\_by\_conversion,

'category\_summary': category\_summary

}

def predict\_sales(self, data):

"""预测销售额"""

live\_data = data['live\_data']

# 特征工程

features = live\_data.copy()

features['hour\_sin'] = np.sin(2 \* np.pi \* features['hour']/24)

features['hour\_cos'] = np.cos(2 \* np.pi \* features['hour']/24)

# 选择特征和目标变量

X = features[['hour\_sin', 'hour\_cos', 'day\_of\_week', 'is\_weekend', 'viewers', 'likes', 'comments', 'shares']]

y = features['revenue']

# 分割训练集和测试集

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# 训练随机森林模型

model = RandomForestRegressor(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# 预测

y\_pred = model.predict(X\_test)

# 评估模型

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

# 获取特征重要性

feature\_importance = pd.DataFrame({

'feature': X.columns,

'importance': model.feature\_importances\_

}).sort\_values('importance', ascending=False)

return {

'model': model,

'mse': mse,

'r2': r2,

'feature\_importance': feature\_importance

}

def generate\_report(self, data, sales\_analysis, audience\_analysis, product\_analysis, prediction):

"""生成分析报告"""

report = {

'overview': {

'total\_revenue': data['daily\_live\_summary']['revenue'].sum(),

'total\_viewers': data['daily\_live\_summary']['viewers'].sum(),

'total\_sales': data['daily\_live\_summary']['sales'].sum(),

'avg\_conversion\_rate': data['daily\_live\_summary']['conversion\_rate'].mean(),

'avg\_ticket\_price': data['daily\_live\_summary']['revenue'].sum() / data['daily\_live\_summary']['sales'].sum() if data['daily\_live\_summary']['sales'].sum() > 0 else 0

},

'sales\_trends': sales\_analysis,

'audience': audience\_analysis,

'products': product\_analysis,

'prediction': prediction

}

return report

def visualize\_report(self, report):

"""可视化分析报告"""

# 创建图表目录

import os

if not os.path.exists('charts'):

os.makedirs('charts')

# 1. 销售额趋势图

plt.figure(figsize=(12, 6))

plt.plot(report['sales\_trends']['daily\_growth']['date'], report['sales\_trends']['daily\_growth']['revenue'], marker='o')

plt.title('每日销售额趋势')

plt.xlabel('日期')

plt.ylabel('销售额')

plt.grid(True, alpha=0.3)

plt.xticks(rotation=45)

plt.tight\_layout()

plt.savefig('charts/daily\_revenue\_trend.png')

# 2. 按小时销售分布图

plt.figure(figsize=(12, 6))

sns.barplot(x='hour', y='revenue', data=report['sales\_trends']['hourly\_sales'])

plt.title('按小时销售额分布')

plt.xlabel('小时')

plt.ylabel('平均销售额')

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.savefig('charts/hourly\_sales.png')

# 3. 观众性别分布

plt.figure(figsize=(8, 8))

plt.pie(

report['audience']['gender\_distribution']['count'],

labels=report['audience']['gender\_distribution']['gender'].map({'male\_users': '男性', 'female\_users': '女性'}),

autopct='%1.1f%%',

startangle=90

)

plt.title('观众性别分布')

plt.tight\_layout()

plt.savefig('charts/gender\_distribution.png')

# 4. 观众年龄分布

plt.figure(figsize=(10, 6))

sns.barplot(x='age\_group', y='percentage', data=report['audience']['age\_distribution'])

plt.title('观众年龄分布')

plt.xlabel('年龄组')

plt.ylabel('百分比(%)')

plt.grid(True, alpha=0.3)

plt.xticks(rotation=45)

plt.tight\_layout()

plt.savefig('charts/age\_distribution.png')

# 5. 商品销售额TOP10

plt.figure(figsize=(12, 6))

sns.barplot(x='revenue', y='product\_name', data=report['products']['top\_products\_by\_revenue'])

plt.title('商品销售额TOP10')

plt.xlabel('销售额')

plt.ylabel('商品名称')

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.savefig('charts/top\_products\_by\_revenue.png')

# 6. 商品转化率TOP10

plt.figure(figsize=(12, 6))

sns.barplot(x='conversion\_rate', y='product\_name', data=report['products']['top\_products\_by\_conversion'])

plt.title('商品转化率TOP10')

plt.xlabel('转化率(%)')

plt.ylabel('商品名称')

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.savefig('charts/top\_products\_by\_conversion.png')

# 7. 特征重要性

plt.figure(figsize=(10, 6))

sns.barplot(x='importance', y='feature', data=report['prediction']['feature\_importance'])

plt.title('销售额预测特征重要性')

plt.xlabel('重要性')

plt.ylabel('特征')

plt.grid(True, alpha=0.3)

plt.tight\_layout()

plt.savefig('charts/feature\_importance.png')

return 'charts'

# 使用示例

if \_\_name\_\_ == "\_\_main\_\_":

# 初始化分析器

analyzer = DouYinEcommerceAnalyzer(api\_key=API\_KEY, secret=SECRET)

# 设置日期范围

start\_date = '2023-05-01'

end\_date = '2023-05-31'

# 获取数据

print("正在获取直播数据...")

live\_data = analyzer.fetch\_live\_stream\_data(start\_date, end\_date)

print("正在获取商品数据...")

product\_data = analyzer.fetch\_product\_data(start\_date, end\_date)

print("正在获取用户数据...")

user\_data = analyzer.fetch\_user\_data(start\_date, end\_date)

# 数据预处理

print("正在进行数据预处理...")

processed\_data = analyzer.preprocess\_data(live\_data, product\_data, user\_data)

# 销售分析

print("正在进行销售分析...")

sales\_analysis = analyzer.analyze\_sales\_trends(processed\_data)

# 观众分析

print("正在进行观众分析...")

audience\_analysis = analyzer.analyze\_audience(processed\_data)

# 商品分析

print("正在进行商品分析...")

product\_analysis = analyzer.analyze\_products(processed\_data)

# 销售额预测

print("正在进行销售额预测...")

prediction = analyzer.predict\_sales(processed\_data)

# 生成报告

print("正在生成分析报告...")

report = analyzer.generate\_report(

processed\_data,

sales\_analysis,

audience\_analysis,

product\_analysis,

prediction

)

# 可视化报告

print("正在生成可视化图表...")

chart\_dir = analyzer.visualize\_report(report)

print(f"分析完成！图表已保存至 {chart\_dir} 目录")

# 打印关键指标

print("\n关键指标概览:")

print(f"总销售额: ¥{report['overview']['total\_revenue']:.2f}")

print(f"总观众数: {report['overview']['total\_viewers']:,}")

print(f"总销量: {report['overview']['total\_sales']:,}")

print(f"平均转化率: {report['overview']['avg\_conversion\_rate']:.2f}%")

print(f"平均客单价: ¥{report['overview']['avg\_ticket\_price']:.2f}")

</code></pre>

</div>

</div>

<div class="grid grid-cols-1 md:grid-cols-2 gap-6 mt-6">

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">抖音开放平台API集成</h3>

<div class="bg-gray-900 text-gray-100 rounded-lg p-4 overflow-x-auto">

<pre><code class="language-python"># 获取抖音开放平台访问令牌

def get\_douyin\_access\_token(app\_id, app\_secret):

url = "https://open.douyin.com/oauth/access\_token/"

params = {

"client\_key": app\_id,

"client\_secret": app\_secret,

"grant\_type": "client\_credentials"

}

response = requests.post(url, params=params)

if response.status\_code == 200:

data = response.json()

return data.get("access\_token")

else:

raise Exception(f"获取访问令牌失败: {response.text}")

# 获取直播数据

def get\_live\_stream\_data(access\_token, start\_time, end\_time, page=1, page\_size=100):

url = "https://open.douyin.com/data/external/live/overview/"

params = {

"access\_token": access\_token,

"start\_time": start\_time,

"end\_time": end\_time,

"page": page,

"page\_size": page\_size

}

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

if response.status\_code == 200:

return response.json()

else:

raise Exception(f"获取直播数据失败: {response.text}")

# 获取商品数据

def get\_product\_data(access\_token, start\_time, end\_time, page=1, page\_size=100):

url = "https://open.douyin.com/data/external/product/overview/"

params = {

"access\_token": access\_token,

"start\_time": start\_time,

"end\_time": end\_time,

"page": page,

"page\_size": page\_size

}

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

if response.status\_code == 200:

return response.json()

else:

raise Exception(f"获取商品数据失败: {response.text}")</code></pre>

</div>

</div>

<div class="bg-white rounded-xl p-6 card-shadow">

<h3 class="text-lg font-semibold mb-4">数据可视化示例</h3>

<div class="bg-gray-900 text-gray-100 rounded-lg p-4 overflow-x-auto">

<pre><code class="language-python"># 销售额趋势图

def plot\_revenue\_trend(data):

plt.figure(figsize=(12, 6))

plt.plot(data['date'], data['revenue'], marker='o', color='#FE2C55')

plt.title('抖音带货销售额趋势')

plt.xlabel('日期')

plt.ylabel('销售额(元)')

plt.grid(True, alpha=0.3)

plt.xticks(rotation=45)

plt.tight\_layout()

return plt

# 商品销售排行

def plot\_top\_products(data, top\_n=10):

top\_products = data.sort\_values('revenue', ascending=False).head(top\_n)

plt.figure(figsize=(12, 6))

sns.barplot(x='revenue', y='product\_name', data=top\_products, color='#222222')

plt.title(f'抖音带货商品销售额TOP{top\_n}')

plt.xlabel('销售额(元)')

plt.ylabel('商品名称')

plt.grid(True, alpha=0.3)

plt.tight\_layout()

return plt

# 观众互动热力图

def plot\_interaction\_heatmap(data):

# 准备数据

pivot\_data = data.pivot\_table(

index='hour',

columns='day\_of\_week',

values='comments',

aggfunc='mean'

)

# 重命名列

pivot\_data.columns = ['周一', '周二', '周三', '周四', '周五', '周六', '周日']

plt.figure(figsize=(12, 8))

sns.heatmap(pivot\_data, cmap='YlOrRd', annot=True, fmt='.1f')

plt.title('抖音直播观众互动热力图')

plt.xlabel('星期')

plt.ylabel('小时')

plt.tight\_layout()

return plt</code></pre>

</div>

</div>

</div>

</section>

</main>

<!-- 页脚 -->

<footer class="bg-secondary text-white py-8">

<div class="container mx-auto px-4">

<div class="grid grid-cols-1 md:grid-cols-3 gap-8">

<div>

<div class="flex items-center space-x-2 mb-4">

<i class="fa fa-line-chart text-primary text-2xl"></i>

<h3 class="text-xl font-bold">抖音带货数据分析</h3>

</div>

<p class="text-gray-400">基于Python和抖音开放平台API构建的数据分析系统，帮助商家洞察带货效果，优化运营策略。</p>

</div>

<div>

<h4 class="text-lg font-semibold mb-4">快速链接</h4>

<ul class="space-y-2">

<li><a href="#dashboard" class="text-gray-400 hover:text-primary transition-colors">数据概览</a></li>

<li><a href="#sales" class="text-gray-400 hover:text-primary transition-colors">销售分析</a></li>

<li><a href="#audience" class="text-gray-400 hover:text-primary transition-colors">观众互动</a></li>

<li><a href="#products" class="text-gray-400 hover:text-primary transition-colors">商品表现</a></li>

<li><a href="#code" class="text-gray-400 hover:text-primary transition-colors">代码实现</a></li>

</ul>

</div>

<div>

<h4 class="text-lg font-semibold mb-4">联系我们</h4>

<ul class="space-y-2">

<li class="flex items-center">

<i class="fa fa-envelope text-primary mr-2"></i>

<a href="mailto:contact@example.com" class="text-gray-400 hover:text-primary transition-colors">contact@example.com</a>

</li>

<li class="flex items-center">

<i class="fa fa-phone text-primary mr-2"></i>

<a href="tel:+861234567890" class="text-gray-400 hover:text-primary transition-colors">+86 123 4567 890</a>

</li>

<li class="flex items-center">

<i class="fa fa-map-marker text-primary mr-2"></i>

<span class="text-gray-400">北京市海淀区中关村科技园区</span>

</li>

</ul>

</div>

</div>

<div class="border-t border-gray-800 mt-8 pt-8 text-center text-gray-500 text-sm">

<p>© 2023 抖音带货数据分析系统 | 版权所有</p>

</div>

</div>

</footer>

<script>

// 图表初始化

document.addEventListener('DOMContentLoaded', function() {

// 销售额趋势图

const salesCtx = document.getElementById('salesChart').getContext('2d');

new Chart(salesCtx, {

type: 'line',

data: {

labels: ['5/1', '5/6', '5/11', '5/16', '5/21', '5/26', '5/31'],

datasets: [{

label: '销售额(元)',

data: [32000, 38000, 42000, 45000, 49000, 52000, 55000],

borderColor: '#FE2C55',

backgroundColor: 'rgba(254, 44, 85, 0.1)',

fill: true,

tension: 0.4

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

display: false

}

},

scales: {

y: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

x: {

grid: {

display: false

}

}

}

}

});

// 销售渠道分布

const channelCtx = document.getElementById('channelChart').getContext('2d');

new Chart(channelCtx, {

type: 'doughnut',

data: {

labels: ['直播', '短视频', '商品橱窗', '搜索推荐'],

datasets: [{

data: [55, 25, 15, 5],

backgroundColor: [

'#FE2C55',

'#00F2EA',

'#222222',

'#FFD700'

],

borderWidth: 0

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

position: 'bottom',

labels: {

boxWidth: 12,

padding: 15

}

}

},

cutout: '70%'

}

});

// 按小时销售分布

const hourlyCtx = document.getElementById('hourlySalesChart').getContext('2d');

new Chart(hourlyCtx, {

type: 'bar',

data: {

labels: ['00:00', '03:00', '06:00', '09:00', '12:00', '15:00', '18:00', '21:00'],

datasets: [{

label: '销售额(元)',

data: [1200, 800, 1500, 3200, 4500, 5200, 4800, 5800],

backgroundColor: '#222222',

borderRadius: 4

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

display: false

}

},

scales: {

y: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

x: {

grid: {

display: false

}

}

}

}

});

// 地域销售TOP10

const regionCtx = document.getElementById('regionChart').getContext('2d');

new Chart(regionCtx, {

type: 'bar',

data: {

labels: ['广东', '江苏', '浙江', '山东', '河南', '四川', '湖北', '福建', '湖南', '北京'],

datasets: [{

label: '销售额(元)',

data: [42000, 35000, 32000, 28000, 25000, 22000, 20000, 18000, 17000, 15000],

backgroundColor: '#00F2EA',

borderRadius: 4

}]

},

options: {

indexAxis: 'y',

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

display: false

}

},

scales: {

x: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

y: {

grid: {

display: false

}

}

}

}

});

// 观众性别分布

const genderCtx = document.getElementById('genderChart').getContext('2d');

new Chart(genderCtx, {

type: 'pie',

data: {

labels: ['男性', '女性'],

datasets: [{

data: [42, 58],

backgroundColor: [

'#3498db',

'#e74c3c'

],

borderWidth: 0

}]

},

options: {

responsive: true,

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

position: 'bottom',

labels: {

boxWidth: 12,

padding: 15

}

}

}

}

});

// 观众年龄分布

const ageCtx = document.getElementById('ageChart').getContext('2d');

new Chart(ageCtx, {

type: 'bar',

data: {

labels: ['18岁以下', '18-24岁', '25-34岁', '35-44岁', '45岁以上'],

datasets: [{

label: '占比(%)',

data: [10, 30, 35, 15, 10],

backgroundColor: '#FE2C55',

borderRadius: 4

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

display: false

}

},

scales: {

y: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

x: {

grid: {

display: false

}

}

}

}

});

// 互动行为分布

const interactionCtx = document.getElementById('interactionChart').getContext('2d');

new Chart(interactionCtx, {

type: 'doughnut',

data: {

labels: ['点赞', '评论', '分享', '转发', '关注'],

datasets: [{

data: [50, 25, 15, 5, 5],

backgroundColor: [

'#FE2C55',

'#00F2EA',

'#222222',

'#FFD700',

'#9b59b6'

],

borderWidth: 0

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

position: 'bottom',

labels: {

boxWidth: 12,

padding: 15

}

}

},

cutout: '70%'

}

});

// 直播时段互动趋势

const interactionTrendCtx = document.getElementById('interactionTrendChart').getContext('2d');

new Chart(interactionTrendCtx, {

type: 'line',

data: {

labels: ['00:00', '03:00', '06:00', '09:00', '12:00', '15:00', '18:00', '21:00'],

datasets: [

{

label: '点赞数',

data: [12000, 8000, 15000, 32000, 45000, 52000, 48000, 58000],

borderColor: '#FE2C55',

backgroundColor: 'rgba(254, 44, 85, 0.1)',

fill: true,

tension: 0.4

},

{

label: '评论数',

data: [2000, 1500, 2500, 5000, 7500, 8500, 8000, 9500],

borderColor: '#00F2EA',

backgroundColor: 'rgba(0, 242, 234, 0.1)',

fill: true,

tension: 0.4

}

]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

position: 'top',

labels: {

boxWidth: 12,

padding: 15

}

}

},

scales: {

y: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

x: {

grid: {

display: false

}

}

}

}

});

// 商品销售额TOP10

const productSalesCtx = document.getElementById('productSalesChart').getContext('2d');

new Chart(productSalesCtx, {

type: 'bar',

data: {

labels: ['智能手表', '口红套装', '蓝牙耳机', '料理锅', 'T恤', '护肤品', '美食礼包', '太阳镜', '健身器材', '扫地机器人'],

datasets: [{

label: '销售额(元)',

data: [85000, 72000, 68000, 52000, 48000, 45000, 38000, 32000, 29000, 25000],

backgroundColor: '#222222',

borderRadius: 4

}]

},

options: {

indexAxis: 'y',

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

display: false

}

},

scales: {

x: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

y: {

grid: {

display: false

}

}

}

}

});

// 商品转化率对比

const conversionCtx = document.getElementById('conversionChart').getContext('2d');

new Chart(conversionCtx, {

type: 'bar',

data: {

labels: ['口红套装', '美食礼包', 'T恤', '太阳镜', '护肤品', '蓝牙耳机', '料理锅', '健身器材', '智能手表', '扫地机器人'],

datasets: [{

label: '转化率(%)',

data: [6.8, 6.5, 5.8, 5.5, 5.2, 4.8, 4.2, 3.9, 3.5, 2.8],

backgroundColor: '#00F2EA',

borderRadius: 4

}]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

display: false

}

},

scales: {

y: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

x: {

grid: {

display: false

}

}

}

}

});

// 商品类别销售分析

const categoryCtx = document.getElementById('categoryChart').getContext('2d');

new Chart(categoryCtx, {

type: 'bar',

data: {

labels: ['美妆个护', '数码家电', '食品饮料', '服装配饰', '家居日用', '运动户外'],

datasets: [

{

label: '销售额(元)',

data: [120000, 200000, 45000, 80000, 60000, 35000],

backgroundColor: '#FE2C55',

borderRadius: 4

},

{

label: '销量',

data: [600, 400, 300, 900, 200, 100],

backgroundColor: '#222222',

borderRadius: 4

}

]

},

options: {

responsive: true,

maintainAspectRatio: false,

plugins: {

legend: {

position: 'top',

labels: {

boxWidth: 12,

padding: 15

}

}

},

scales: {

y: {

beginAtZero: true,

grid: {

color: 'rgba(0, 0, 0, 0.05)'

}

},

x: {

grid: {

display: false

}

}

}

}

});

});

// 平滑滚动

document.querySelectorAll('a[href^="#"]').forEach(anchor => {

anchor.addEventListener('click', function (e) {

e.preventDefault();

document.querySelector(this.getAttribute('href')).scrollIntoView({

behavior: 'smooth'

});

});

});

</script>

</body>

</html>