Startup Decision Support System

Arnav Jain

Sachin Kumar Soni

Dhruva Sharma

Overview

The Startup Decision Support System (SDSS) is an innovative tool designed to assist entrepreneurs and leadership teams in making informed decisions by analyzing key performance indicators (KPIs) relevant to their startup. Built using advanced data analytics and artificial intelligence (AI) technologies, the system provides a comprehensive dashboard that visualizes business metrics and trends over time. This project leverages cutting-edge natural language processing (NLP) models and retrieval-augmented generation techniques to offer insights based on real-time industry data.

Key Features

1. KPI Selection and Visualization

The SDSS allows users to select up to 12 KPIs that are most relevant to their startup's performance. Some of the KPIs available for selection include:

- Customer Acquisition Cost (CAC)
- Burn Rate
- Monthly Active Users (MAU)
- Revenue per Employee
- Total Gross Sales

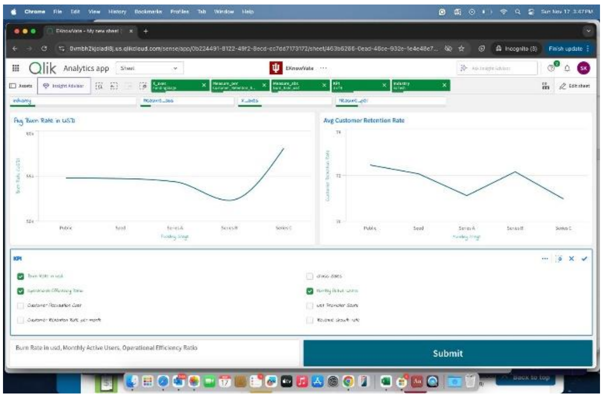
Once selected, these KPIs are plotted on an interactive dashboard, displaying trends over time. The dashboard includes two main charts:

- **Absolute Numbers Chart**: Displays raw numerical values such as total revenue or CAC.
- **Percentage Change Chart**: Shows the percentage change in metrics like revenue or burn rate over time.

These visualizations help users understand how their startup is performing relative to its age, funding stage, and other critical factors.

2. Time Series Analysis

The system provides time series analysis, allowing users to track changes in their selected KPIs over different time periods. This feature enables startups to monitor performance trends, identify growth opportunities, and spot potential risks early on.

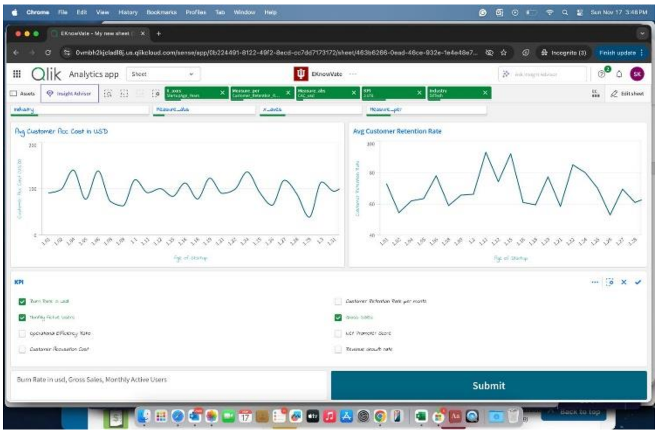


KPI

- ☑ Burn Rate in usd
- ☑ Operational Efficiency Ratio
- □ Customer Acquisition Cost
- □ Customer Retention Rate per month
- □ Gross Sales
- ☑ Monthly Active Users
- □ Net Promoter Score
- □ Revenue Growth rate

Burn Rate in usd, Monthly Active Users, Operational Efficiency Ratio

Submit

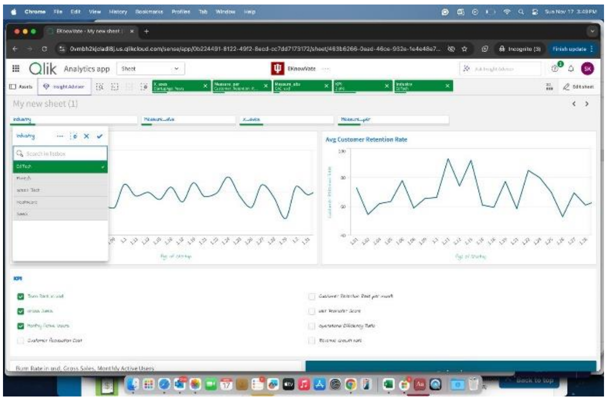


KPI

- ☑ Burn Rate in usd
- ☑ Monthly Active Users
- □ Operational Efficiency Ratio
- □ Customer Acquisition Cost
- □ Customer Retention Rate per month
- ☑ Gross Sales
- □ Net Promoter Score
- □ Revenue Growth rate

Burn Rate in usd, Gross Sales, Monthly Active Users

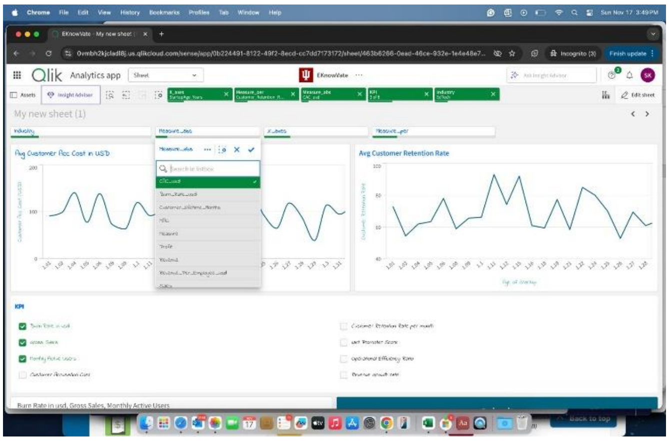
Submit



KPI

- ☑ Burn Rate in usd
- ☑ Gross Sales
- ☑ Monthly Active Users
- □ Customer Acquisition Cost
- □ Customer Retention Rate per month
- □ Net Promoter Score
- □ Operational Efficiency Ratio
- □ Revenue Growth rate

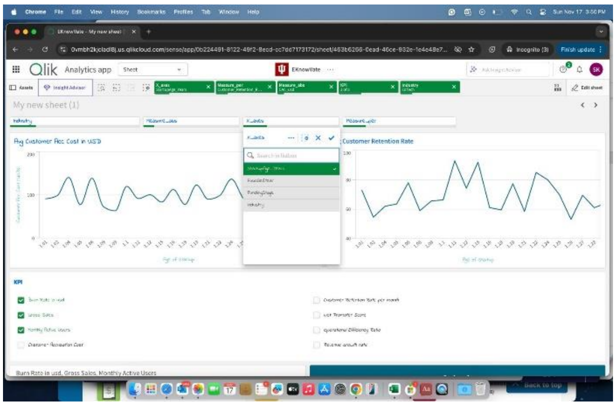
Burn Rate in usd, Gross Sales, Monthly Active Users



KPI

- ☑ Burn Rate in usd
- ☑ Gross Sales
- ☑ Monthly Active Users
- □ Customer Acquisition Cost
- □ Customer Retention Rate per month
- □ Net Promoter Score
- □ Operational Efficiency Ratio
- □ Revenue Growth rate

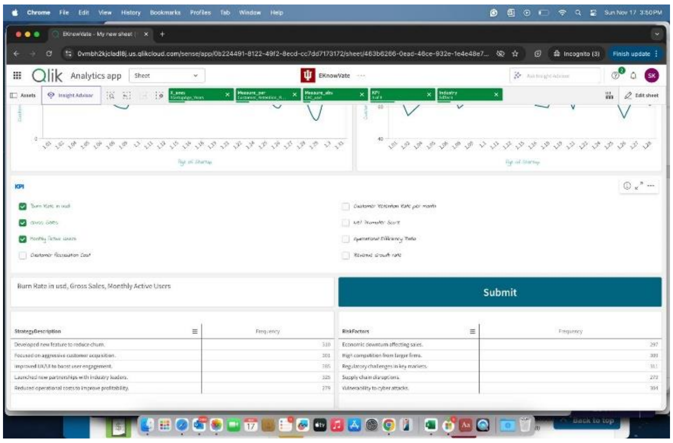
Burn Rate in usd, Gross Sales, Monthly Active Users



KPI

- ☑ Burn Rate in usd
- ☑ Gross Sales
- ☑ Monthly Active Users
- □ Customer Acquisition Cost
- □ Customer Retention Rate per month
- □ Net Promoter Score
- □ Operational Efficiency Ratio
- □ Revenue Growth rate

Burn Rate in usd, Gross Sales, Monthly Active Users



KPI

- ☑ Burn Rate in usd
- ☑ Gross Sales
- ☑ Monthly Active Users
- □ Customer Acquisition Cost
- □ Customer Retention Rate per month
- □ Net Promoter Score
- □ Operational Efficiency Ratio
- □ Revenue Growth rate

Burn Rate in usd, Gross Sales, Monthly Active Users

Submit

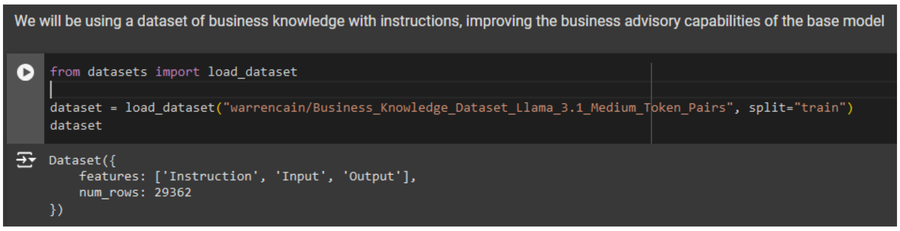
StrategyDescription	Frequency
Developed new feature to reduce churn.	310
Focused on aggressive customer acquisition.	301
Improved UX/UI to boost user engagement.	285
Launched new partnerships with industry leaders.	325
Reduced operational costs to improve profitability.	279

RiskFactors	Frequency
Economic downturn affecting sales.	297
High competition from larger firms.	300
Regulatory challenges in key markets.	311
Supply chain disruptions.	272
Vulnerability to cyber attacks.	304

3. AI-Powered Insights

At the core of the SDSS is a powerful AI engine based on Google's Gemma 2 billion parameter large language model. This model has been fine-tuned on a business knowledge dataset using a cost-efficient quantized approach with the "Bits and Bytes" library. The AI engine is capable of:

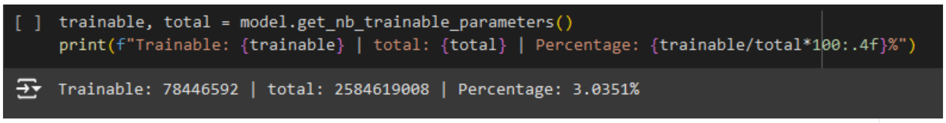
- Interpreting user inputs (prompts) to understand the specific needs of the startup.
- Running vector searches against a database of business news articles and industry data.
- Providing relevant insights based on real-time data while maintaining the context and sentiment of the original prompt.



We will be using a dataset of business knowledge with instructions, improving the business advisory capabilities of the base model

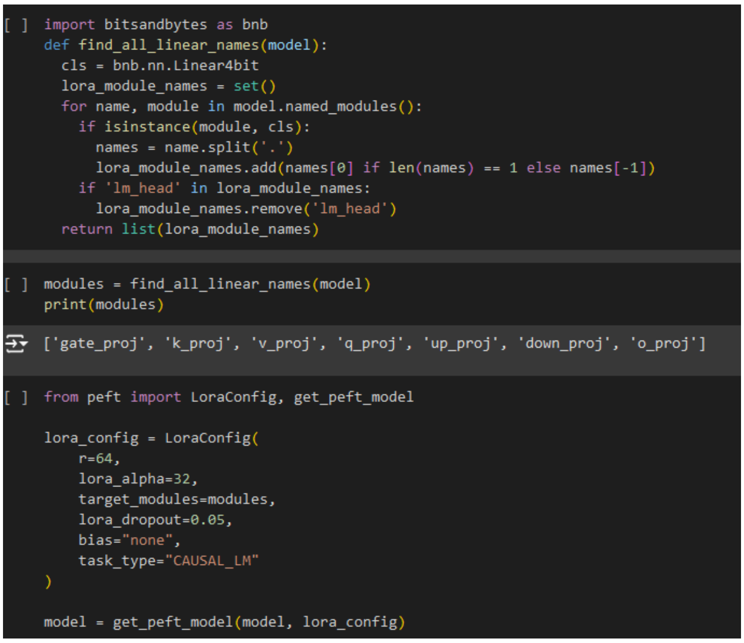
```
from datasets import load_dataset

dataset = load_dataset("warrencain/Business_Knowledge_Dataset_Llama_3.1_Medium_Token_Pairs", split="train")
dataset
```

```
Dataset({
    features: ['Instruction', 'Input', 'Output'],
    num_rows: 29362
})
```



```
trainable, total = model.get_nb_trainable_parameters()
print(f"Trainable: {trainable} | total: {total} | Percentage: {trainable/total*100:.4f}%")
```

```
Trainable: 78446592 | total: 2584619008 | Percentage: 3.0351%
```



```
import bitsandbytes as bnb
def find_all_linear_names(model):
  cls = bnb.nn.Linear4bit
  lora_module_names = set()
  for name, module in model.named_modules():
    if isinstance(module, cls):
      names = name.split('.')
      lora_module_names.add(names[0] if len(names) == 1 else names[-1])
    if 'lm_head' in lora_module_names:
      lora_module_names.remove('lm_head')
  return list(lora_module_names)
```

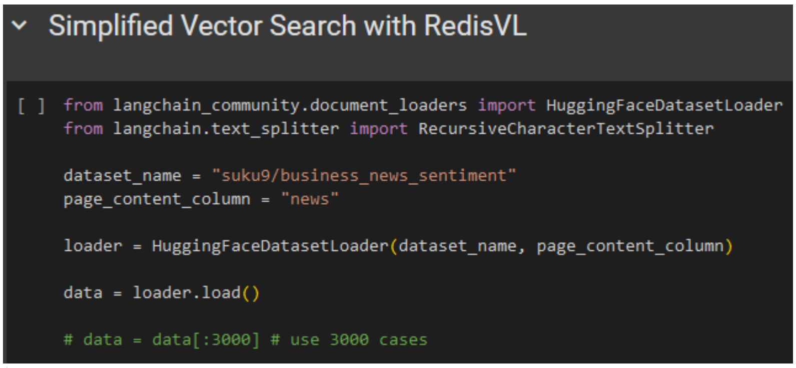
```
modules = find_all_linear_names(model)
print(modules)
```

```
['gate_proj', 'k_proj', 'v_proj', 'q_proj', 'up_proj', 'down_proj', 'o_proj']
```

```
from peft import LoraConfig, get_peft_model

lora_config = LoraConfig(
    r=64,
    lora_alpha=32,
    target_modules=modules,
    lora_dropout=0.05,
    bias="none",
    task_type="CAUSAL_LM"
)

model = get_peft_model(model, lora_config)
```

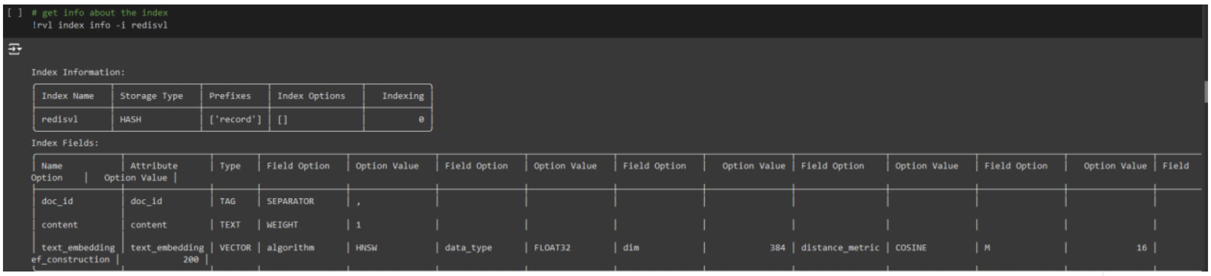
4. Retrieval-Augmented Generation (RAG)

The system employs retrieval-augmented generation (RAG) to enhance its decision-making capabilities. Using a Redis Vector database, SDSS can fetch relevant business news and data that align with the user's input. This ensures that the insights provided are not only accurate but also contextualized within current industry trends.



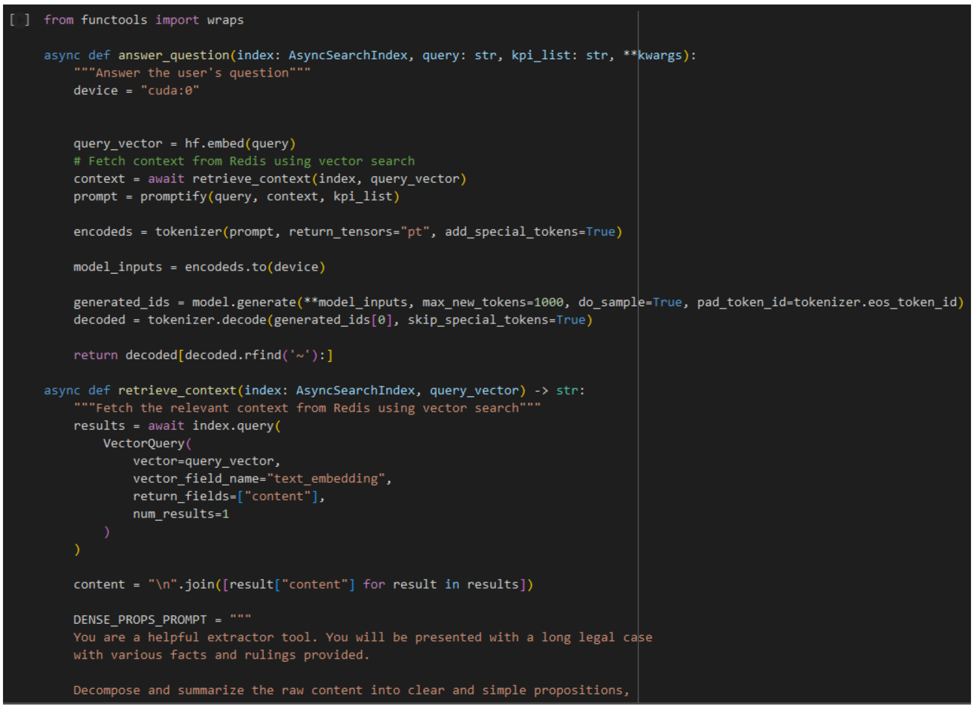
Simplified Vector Search with RedisVL

```
from langchain_community.document_loaders import HuggingFaceDatasetLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter

dataset_name = "suku9/business_news_sentiment"
page_content_column = "news"

loader = HuggingFaceDatasetLoader(dataset_name, page_content_column)

data = loader.load()

# data = data[:3000] # use 3000 cases
```



```
# get info about the index
!rvl index info -i redisvl
```

```
Index Information:
╭──────────────┬────────────────┬────────────┬─────────────────┬────────────╮
│ Index Name   │ Storage Type   │ Prefixes   │ Index Options   │   Indexing │
├──────────────┼────────────────┼────────────┼─────────────────┼────────────┤
│ redisvl      │ HASH           │ ['record'] │ []              │          0 │
╰──────────────┴────────────────┴────────────┴─────────────────┴────────────╯
Index Fields:
╭────────────────┬────────────────┬────────┬────────────────┬────────────────┬────────────────┬────────────────┬────────────────┬────────────────┬─────────────────┬────────────────┬────────────────┬────────────────┬─────────
│ Name           │ Attribute      │ Type   │ Field Option   │ Option Value   │ Field Option   │ Option Value   │ Field Option   │   Option Value │ Field Option    │ Option Value   │ Field Option   │   Option Value │ Field
Option      │    Option Value │
├────────────────┼────────────────┼────────┼────────────────┼────────────────┼────────────────┼────────────────┼────────────────┼────────────────┼─────────────────┼────────────────┼────────────────┼────────────────┼─────────
│ doc_id         │ doc_id         │ TAG    │ SEPARATOR      │ ,              │                │                │                │                │                 │                │                │                │
│ content        │ content        │ TEXT   │ WEIGHT         │ 1              │                │                │                │                │                 │                │                │                │
│ text_embedding │ text_embedding │ VECTOR │ algorithm      │ HNSW           │ data_type      │ FLOAT32        │ dim            │            384 │ distance_metric │ COSINE         │ M              │             16 │
ef_construction │             200 │
```



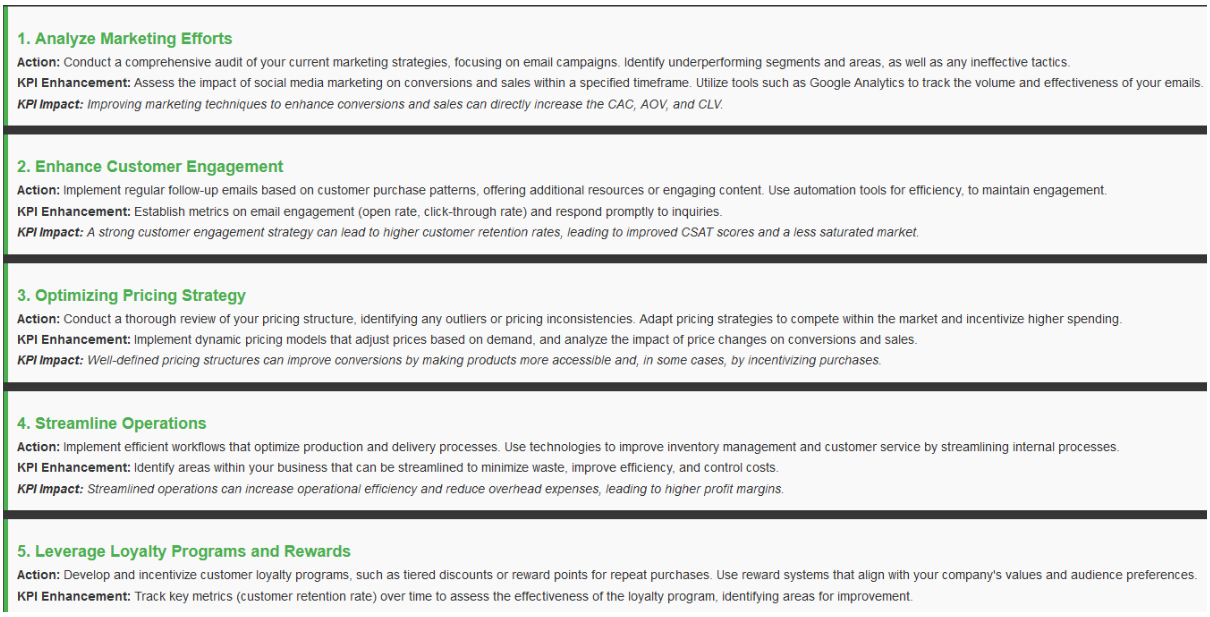
```
from functools import wraps

async def answer_question(index: AsyncSearchIndex, query: str, kpi_list: str, **kwargs):
    """Answer the user's question"""
    device = "cuda:0"


    query_vector = hf.embed(query)
    # Fetch context from Redis using vector search
    context = await retrieve_context(index, query_vector)
    prompt = promptify(query, context, kpi_list)

    encodeds = tokenizer(prompt, return_tensors="pt", add_special_tokens=True)

    model_inputs = encodeds.to(device)

    generated_ids = model.generate(**model_inputs, max_new_tokens=1000, do_sample=True, pad_token_id=tokenizer.eos_token_id)
    decoded = tokenizer.decode(generated_ids[0], skip_special_tokens=True)

    return decoded[decoded.rfind('~'):]

async def retrieve_context(index: AsyncSearchIndex, query_vector) -> str:
    """Fetch the relevant context from Redis using vector search"""
    results = await index.query(
        VectorQuery(
            vector=query_vector,
            vector_field_name="text_embedding",
            return_fields=["content"],
            num_results=1
        )
    )

    content = "\n".join([result["content"] for result in results])

    DENSE_PROPS_PROMPT = """
    You are a helpful extractor tool. You will be presented with a long legal case
    with various facts and rulings provided.

    Decompose and summarize the raw content into clear and simple propositions,
```

Use Case Example

An entrepreneur might use SDSS to assess their startup's burn rate in relation to industry benchmarks. By selecting "Burn Rate" as one of the KPIs, they can see how their company's burn rate has changed over time and compare it with similar startups at different stages of funding. Additionally, by inputting specific queries about market conditions or competitor performance, they can receive AI-generated insights that help them make strategic decisions about resource allocation or fundraising efforts.

Conclusion

The Startup Decision Support System is an essential tool for any entrepreneur or leadership team looking to make data-driven decisions. By combining advanced AI models with real-time industry data, SDSS provides actionable insights that can help startups navigate complex business environments more effectively.



1. Analyze Marketing Efforts

Action: Conduct a comprehensive audit of your current marketing strategies, focusing on email campaigns. Identify underperforming segments and areas, as well as any ineffective tactics.

KPI Enhancement: Assess the impact of social media marketing on conversions and sales within a specified timeframe. Utilize tools such as Google Analytics to track the volume and effectiveness of your emails.

KPI Impact: *Improving marketing techniques to enhance conversions and sales can directly increase the CAC, AOV, and CLV.*

2. Enhance Customer Engagement

Action: Implement regular follow-up emails based on customer purchase patterns, offering additional resources or engaging content. Use automation tools for efficiency, to maintain engagement.

KPI Enhancement: Establish metrics on email engagement (open rate, click-through rate) and respond promptly to inquiries.

KPI Impact: *A strong customer engagement strategy can lead to higher customer retention rates, leading to improved CSAT scores and a less saturated market.*

3. Optimizing Pricing Strategy

Action: Conduct a thorough review of your pricing structure, identifying any outliers or pricing inconsistencies. Adapt pricing strategies to compete within the market and incentivize higher spending.

KPI Enhancement: Implement dynamic pricing models that adjust prices based on demand, and analyze the impact of price changes on conversions and sales.

KPI Impact: *Well-defined pricing structures can improve conversions by making products more accessible and, in some cases, by incentivizing purchases.*

4. Streamline Operations

Action: Implement efficient workflows that optimize production and delivery processes. Use technologies to improve inventory management and customer service by streamlining internal processes.

KPI Enhancement: Identify areas within your business that can be streamlined to minimize waste, improve efficiency, and control costs.

KPI Impact: *Streamlined operations can increase operational efficiency and reduce overhead expenses, leading to higher profit margins.*

5. Leverage Loyalty Programs and Rewards

Action: Develop and incentivize customer loyalty programs, such as tiered discounts or reward points for repeat purchases. Use reward systems that align with your company's values and audience preferences.

KPI Enhancement: Track key metrics (customer retention rate) over time to assess the effectiveness of the loyalty program, identifying areas for improvement.