

# B2B 고객 제품 수요량 예측 기반 재고 최적화

데이터 분석 및 예측을 활용한 발주 시스템



# INDEX

---

1. 프로젝트 소개
2. 문제 분석 및 목표 설정
3. 주요 코드 – 메인코드
4. 주요 코드 – 시각화 코드
5. 기대 효과





# 프로젝트 팀 소개



문시크42 (MoonChic42)



팀장 : 정태영

기획, 데이터 수집, 알고리즘 테스트, 서포트



팀원 : 문성준, 김병지, 조지형, 조준행

데이터 수집, 데이터 전처리, 알고리즘 구현, 알고리즘 테스트



# 문제 인식 및 목표 설정



**목표 달성**

재고 비용 20% 절감 & 데이터 시각화

**맞춤형 솔루션**

Prophet 모델 기반 재고데이터 활용 수요 예측

**문제 인식**

130개의 제품으로 인하여 재고관리 및  
불규칙한 입고 관리에 어려움이 있음

# 메인코드 알고리즘 기획

## 리드타임 분석

공급업체별 배송 시간  
데이터 수집

## 재고 최적화

안전재고 수준 자동 계산

## 수요 예측

제품별, 고객별  
수요 패턴 분석

## VIP 고객 관리

최대 구매자 및 최대 매출제품  
TOP 10 리스트 관리

## 통합 데이터 분석



# 예측 시각화 코드 알고리즘 기획



## 현재 데이터 분석

현재까지의 판매 패턴과 트렌드 파악



## 단기 예측(1-3개월)

95% 신뢰구간 내 정확도 확보



## 중기 예측(4-6개월)

시장 변화 요인 반영한 시나리오 분석



## 장기 예측(12개월+)

계절성 요인 고려한 변동성 예측





# 주요 코드 소개 – 메인 코드

## Prophet 모델 구현

Facebook(메타)에서 개발한 시계열 예측 알고리즘으로 추세(Trend), 계절성(Seasonality), 이벤트 효과(Holiday Effects)를 자동으로 모델링 하며, 자동화된 데이터 전처리 기능을 제공

## 예측 알고리즘

과거 6년간의 판매 데이터 학습  
12개월 수요 예측 기능 구현  
발주 후리드타임 7일 계산

## 알고리즘 모델 코드

```
if target_product.strip().lower() == "all":
    product_list = df["product"].unique()
    print("\n✓ 전체 제품 데이터를 사용하여 예측을 진행합니다.", end="")
    # 초기 메시지 출력

else:
    product_list = [target_product]

total_products = len(product_list) # 전체 제품 개수

for idx, product in enumerate(product_list, start=1):
    progress = (idx / total_products) * 100 # 진행률 계산
    bike_icon = next(bike_frames) # 자전거 모션 업데이트

    # '\r'를 사용해 한 줄에서 업데이트
    print(f"\r✓ 전체 제품 데이터를 사용하여 예측을 진행합니다.
    ({progress:.1f}% 완료) {bike_icon}", end="")
    sys.stdout.flush() # 즉시 출력 반영

    # Prophet 모델 학습
    product_df = df[df["product"] == product]
    model = Prophet()
    model.fit(product_df)
    future = model.make_future_dataframe(periods=30)
    forecast = model.predict(future)
    forecast_results[product] = forecast

    time.sleep(0.1)

# 진행 속도 조절
# 완료 후 개행하여 최종 메시지 출력
print("\n✓ 모든 제품의 예측이 완료되었습니다!")
```

# 주요 코드 소개 - 시각화



## 데이터 로드 및 전처리

Pandas를 활용한 데이터  
정제 및 구조화

## 알고리즘 구현

Prophet 모델 기반  
2011~2016 데이터셋 학습

## 기본 시각화 구현

Matplotlib, Tabulate 으로  
기초 트렌드 그래프 생성

## 성능 최적화

대용량 데이터 처리를 위한  
렌더링 최적화



# 주요 코드 소개 — 시각화 예측 그래프

## Metplotlib 시각화 구현

Facebook(메타)에서 개발한 시계열 예측 알고리즘과 metplotlib 모듈을 활용하여 온라인 채널 및 시계열 예측 그래프를 제작

## 예측 알고리즘

과거 데이터 데이터 타입 형식  
1개월 단위 기준 12개월 동안 데이터 예측  
Metplotlib를 활용하여 그래프 시각화

## 알고리즘 모델 코드

```
import pandas as pd
import matplotlib.pyplot as plt
from prophet import Prophet
```

```
# CSV 파일 로드
file_path = "sales_data.csv"
df = pd.read_csv(file_path)
```

```
# 날짜 변환
df['Date'] = pd.to_datetime(df['Date'])

# 특정 제품 입력
product_name = input("예측할 제품명을 입력하세요: ")

# 특정 제품만 필터링
df_product = df[df['Product'] == product_name]

# Prophet이 요구하는 컬럼명으로 변경
df_product = df_product.rename(columns={'Date': 'ds', 'Revenue': 'y'})
df_product = df_product[['ds', 'y']].groupby('ds').sum().reset_index()
```

```
# Prophet 모델 학습
model = Prophet()
model.fit(df_product)
```

```
# 미래 예측 데이터프레임 생성 (3년 예측)
future = model.make_future_dataframe(periods=12, freq='M')

# 12개월 예측
forecast = model.predict(future)
```

```
# 결과 시각화
fig, ax = plt.subplots(figsize=(10, 5))
ax.plot(df_product['ds'], df_product['y'], label='Actual Revenue')
ax.plot(forecast['ds'], forecast['yhat'], label='Predicted Revenue',
        linestyle='dashed')
ax.set_title(f'Revenue Forecast for {product_name} (Next 1 Years)')
ax.legend()plt.show()
```

# 주요 코드 소개 — 시각화 TOP 10 리스트

## tabulate 모델 구현

파이썬에서 리스트를 표로 표현할 수  
있는 tabulate 모듈을 활용하여  
다양한 테이블 형식의 표를 제작

## 알고리즘 모델 코드

```
!pip install tabulate
import pandas as pd
from tabulate import tabulate # 표 형태로 출력
```

```
# CSV 파일 불러오기
file_path = "sales_data.csv"
df = pd.read_csv(file_path)
```

```
# 결측값 제거 (NaN 값 삭제)
df = df.dropna()
```

```
# 🏆 VIP 고객 찾기 (매출 기준 상위 5명)
vip_customers = df.nlargest(5, 'Revenue')
```

```
# 📈 가장 많이 판매된 제품 TOP 5 찾기
top_5_products = df['Product'].value_counts().head(5).reset_index()
top_5_products.columns = ['Product', 'Sales Count']
```

# ✔ 결과 출력

```
print("\n🏆 VIP 고객 정보 (상위 5명):")
print(tabulate(vip_customers[['Customer_Age',
'Customer_Gender', 'Country', 'Revenue']], headers='keys',
tablefmt='pretty'))
```

# 📈 가장 많이 판매된 제품 TOP 5:

```
top_5_products.index = range(1, len(top_5_products) + 1)
```

# 인덱스를 1부터 시작하도록 설정

```
print("\n📈 가장 많이 판매된 제품 TOP 5:")
print(tabulate(top_5_products, headers='keys',
tablefmt='pretty'))
```

## 데이터 내용

상위 5명의 VIP 고객 정보  
가장 많이 판매된 제품 TOP 5



# B2B Demand Forecasting



## Step 1

Data collection, into passion data  
essoucsiou nrent for clou demand  
cloud data sttorage, and streams.  
Into collecting but thim B2B in ssuraces  
and fevolysis and statistical models.



## Step 3

Data collection and In demand  
forecasting In frerohcom Data analysts,  
Data analysts and forecating, and  
recreshants, and statistical models.  
Data analysis and forecstasts, and  
analysis cond itenend relch rendifes.



## Implementation

The causee in plantly usiets of demand  
cheresare coreme atungsing arovuceshes  
for an emproment if the :and fonacs, key  
demous of thecturing alroul demand  
an layiogrererhs the tive and intepronato  
and morerematies.

# 전체 프로세스 로드맵



## 데이터 수집

판매 이력, 고객 정보, 시장 데이터 확보



## 데이터 전처리

이상치 제거 및 결측값 처리



## 모델 학습

Prophet 알고리즘 적용 및 모델학습 및 최적화



## 시스템 구현

실시간 모니터링 대시보드 개발



# 프로젝트 마무리 및 기대효과

## 25%

### 재고 비용 절감

과잉재고 감소 및 저장 비용 최소화

## 15%

### 주문 취소율 감소

재고 부족으로 인한 고객 이탈 방지

## 30%

### 운영 효율성 향상

자동화된 발주 시스템으로  
인력 비용 절감



감사합니다

MoonChic42

