1　多様なコンテンツが混在する技術文書

1.1　□ 目次

1.2　□ 数式・化学式セクション

1.2.1　数学的表現

基本数式　線形代数における固有値問題：**Ax = λx**
　ここで、A は n×n 行列、λ は固有値、x は固有ベクトルを表す。

微積分　関数 f(x) = x² + 2x + 1 の導関数：**f'(x) = 2x + 2**
　定積分の計算：**∫□¹ x² dx = [x³/3]□¹ = 1/3**

統計・確率　正規分布の確率密度関数：**f(x) = (1/√(2πσ²)) × e^(-(x-μ)²/(2σ²))**
　ベイズの定理：**P(A|B) = P(B|A) × P(A) / P(B)**

1.2.2　化学式・分子構造

有機化合物

- **グルコース**: C□H□□O□
- **エタノール**: C□H□OH

- **アスピリン**: C□H□O□

化学反応式　燃焼反応：**CH□ + 2O□ → CO□ + 2H□O**
　酸塩基反応：**HCl + NaOH → NaCl + H□O**

物理化学定数

定数	記号	値	単位
アボガドロ数	N_a	6.022 × 10²³	mol□¹
光速	c	2.998 × 10□	m/s
プランク定数	h	6.626 × 10□³□	J□s
重力加速度	g	9.807	m/s²

1.3 □ プログラミングコード

1.3.1 Python 実装例

```
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

class DataProcessor:
    """ データ前処理を行うクラス"""

    def __init__(self, scaling=True, test_size=0.2):
        self.scaling = scaling
        self.test_size = test_size
        self.scaler = StandardScaler() if scaling else None

    def preprocess(self, df, target_column):
        """
        データの前処理を実行

        Args:
            df (pd.DataFrame): 入力データフレーム
            target_column (str): 目的変数のカラム名

        Returns:
            tuple: (X_train, X_test, y_train, y_test)
        """
        # 欠損値処理
        df_clean = df.dropna()

        # 特徴量と目的変数の分離
        X = df_clean.drop(columns=[target_column])
        y = df_clean[target_column]

        # 学習・テストデータ分割
        X_train, X_test, y_train, y_test = train_test_split(
            X, y, test_size=self.test_size, random_state=42
        )

        # 標準化
        if self.scaling:
            X_train = self.scaler.fit_transform(X_train)
            X_test = self.scaler.transform(X_test)

        return X_train, X_test, y_train, y_test

# 使用例
processor = DataProcessor(scaling=True, test_size=0.3)
X_train, X_test, y_train, y_test = processor.preprocess(data, 'target')
```

データ処理パイプライン

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

def evaluate_model(model, X_test, y_test, model_name="Model"):
    """
    モデルの評価を行う関数
    """
    y_pred = model.predict(X_test)

    # 評価指標計算
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)

    print(f"=== {model_name} 評価結果 ===")
    print(f"MSE:  {mse:.4f}")
    print(f"RMSE: {rmse:.4f}")
    print(f"R²:   {r2:.4f}")

    # 予測 vs 実際値のプロット
    plt.figure(figsize=(8, 6))
    plt.scatter(y_test, y_pred, alpha=0.6)
    plt.plot([y_test.min(), y_test.max()],
             [y_test.min(), y_test.max()], 'r--', lw=2)
    plt.xlabel('実際値')
    plt.ylabel('予測値')
    plt.title(f'{model_name} - 予測精度')
    plt.show()

    return {'mse': mse, 'rmse': rmse, 'r2': r2}
```

機械学習モデル

1.3.2 JavaScript (React)

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';

const DataVisualization = ({ dataEndpoint }) => {
  const [data, setData] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await axios.get(dataEndpoint);
        setData(response.data);
      } catch (err) {
        setError(err.message);
      } finally {
```

```
        setLoading(false);
      }
    };

    fetchData();
  }, [dataEndpoint]);

  const processData = (rawData) => {
    return rawData
      .filter(item => item.value > 0)
      .map(item => ({
        ...item,
        normalized: item.value / Math.max(...rawData.map(d => d.value))
      }))
      .sort((a, b) => b.value - a.value);
  };

  if (loading) return <div className="loading">データ読み込み中...</div>;
  if (error) return <div className="error">エラー: {error}</div>;

  const processedData = processData(data);

  return (
    <div className="data-visualization">
      <h2>データ可視化</h2>
      <div className="chart-container">
        {processedData.map((item, index) => (
          <div key={item.id} className="bar-item">
            <span className="label">{item.name}</span>
            <div
              className="bar"
              style={{ width: `${item.normalized * 100}%` }}
            >
              {item.value}
            </div>
          </div>
        ))}
      </div>
    </div>
  );
};

export default DataVisualization;
```

1.3.3 SQL クエリ

```
-- 複雑な分析クエリの例
WITH monthly_sales AS (
    SELECT
        DATE_TRUNC('month', order_date) AS month,
        product_category,
        SUM(amount) AS total_sales,
```

```
        COUNT(*) AS order_count,
        AVG(amount) AS avg_order_value
    FROM orders o
    JOIN products p ON o.product_id = p.id
    WHERE order_date >= '2024-01-01'
    GROUP BY DATE_TRUNC('month', order_date), product_category
),
category_ranking AS (
    SELECT
        month,
        product_category,
        total_sales,
        ROW_NUMBER() OVER (
            PARTITION BY month
            ORDER BY total_sales DESC
        ) AS sales_rank
    FROM monthly_sales
)
SELECT
    month,
    product_category,
    total_sales,
    sales_rank,
    LAG(total_sales) OVER (
        PARTITION BY product_category
        ORDER BY month
    ) AS prev_month_sales,
    ROUND(
        (total_sales - LAG(total_sales) OVER (
            PARTITION BY product_category
            ORDER BY month
        )) / LAG(total_sales) OVER (
            PARTITION BY product_category
            ORDER BY month
        ) * 100, 2
    ) AS growth_rate_pct
FROM category_ranking
WHERE sales_rank <= 5
ORDER BY month DESC, sales_rank ASC;
```

1.4 □ 多言語混在テキスト

1.4.1 日英混在文書

概要: This document demonstrates 多言語対応 in PDF conversion systems. 特に Japanese と English が混在する technical documentation において、proper parsing and structure preservation が重要である。

Key Challenges:

1. **Character encoding issues**
    - UTF-8 vs Shift-JIS compatibility
    - 特殊文字 (□, □, ♪) の handling

- Emoji support: □□□□

2. **Typography differences**
    - 英語: Proportional fonts (Arial, Helvetica)
    - 日本語: Fixed-width fonts (ゴシック, 明朝)
    - Mixed text: バランスの取れた font selection
3. **Reading direction complexity**
    - Horizontal: left-to-right (English, 横書き日本語)
    - Vertical: top-to-bottom, right-to-left (縦書き日本語)

1.4.2 中国語・韓国語サンプル

⊠体中文示例　机器学⊠在文档⊠理中的⊠用
⊠代文档⊠理系⊠广泛采用深度学⊠技⊠来提高⊠⊠精度。主要包括：

- **卷⊠神⊠网⊠** (CNN): 用于⊠像特征提取
- **循⊠神⊠网⊠** (RNN): ⊠理序列数据
- **注意力机制**: 改⊠⊠序列⊠理能力

한국어예시　문서변환시스템의기술적과제
PDF 에서마크다운으로의변환과정에서다음과같은기술적어려움이있습니다:

1. **레이아웃인식**: 복잡한다단구조
2. **표구조파악**: 병합된셀처리
3. **폰트정보보존**: 다양한서체지원

1.4.3 특수문자및기호

수학기호　□x □ □, □y □ □ such that x ≤ y

화폐단위

- USD: $1,234.56
- EUR: €1.234,56
- JPY: ¥123,456
- GBP: £1,234.56
- KRW: ₩1,234,567

단위기호

- 길이: mm, cm, m, km, inch (″), feet (′)
- 무게: mg, g, kg, t, oz, lb
- 온도: °C, °F, K
- 각도: °, ′, ″, rad

1.5 □ 복잡한표구조

1.5.1 다층헤더테이블

지역			2024 년 분기별매출 (백만원)				전년대비	
			Q1	Q2	Q3	Q4	증감액	증감률
아시아	**태평양**	한국	1,200	1,350	1,280	1,420	+180	+15.2%
		일본	2,100	2,280	2,150	2,380	+290	+14.1%
		중국	3,800	4,200	4,100	4,500	+650	+16.9%
		기타	890	920	980	1,050	+140	+15.8%
		소계	**7,990**	**8,750**	**8,510**	**9,350**	**+1,260**	**+15.9%**
북미		미국	5,200	5,800	5,600	6,100	+880	+16.9%
		캐나다	1,100	1,200	1,180	1,280	+160	+14.5%
		소계	**6,300**	**7,000**	**6,780**	**7,380**	**+1,040**	**+16.5%**
유럽		독일	1,800	1,950	1,900	2,080	+230	+12.4%
		프랑스	1,200	1,300	1,250	1,380	+180	+15.0%
		영국	1,500	1,620	1,580	1,720	+200	+13.3%
		기타	800	850	820	900	+110	+13.8%
		소계	**5,300**	**5,720**	**5,550**	**6,080**	**+720**	**+13.6%**
			총계	**19,590**	**21,470**	**20,840**	**22,810**	**+3,020**

1.5.2 불규칙한셀병합테이블

프로젝트	담당팀		예산 (백만원)		진행률	상태
	개발	운영	2024 년	2025 년		
AI 플랫폼	15 명	8 명	1,200	800	85%	□ 정상
클라우드이전	12 명	15 명	800	400	62%	□ 주의
보안강화	8 명	12 명	600	300	78%	□ 정상
모바일앱	10 명	5 명	500	200	45%	□ 지연

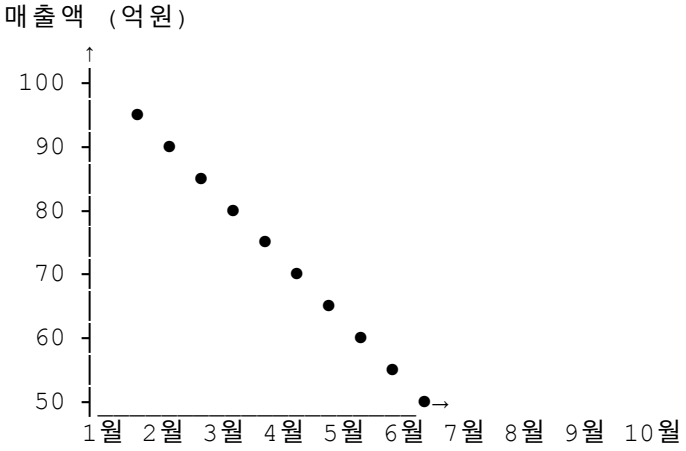
1.5.3 계산식이포함된테이블

상품	수량	단가	소계	할인율	할인액	최종금액
제품 A	100	1,500	150,000	10%	15,000	135,000
제품 B	75	2,200	165,000	15%	24,750	140,250
제품 C	50	3,000	150,000	5%	7,500	142,500
제품 D	200	800	160,000	20%	32,000	128,000
		합계	**625,000**		**79,250**	**545,750**
				부가세 (10%)		**54,575**
				총결제금액		**600,325**

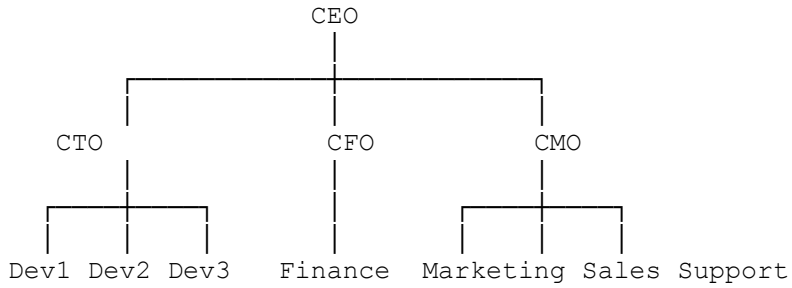
1.6 □ 도표·차트

1.6.1 ASCII 아트차트

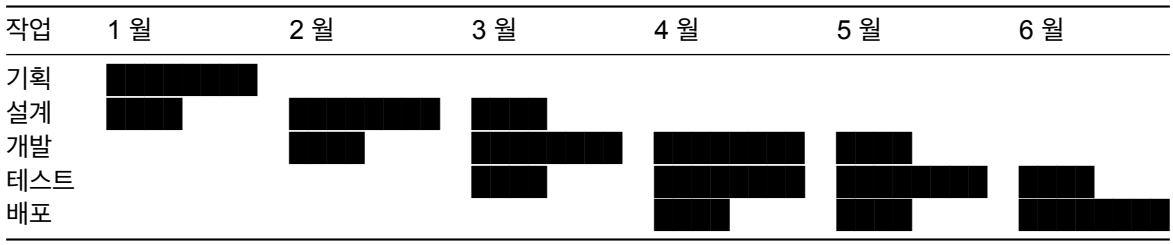
월별매출추이



조직도



1.6.2 가나드차트 (간트차트)



1.7 □ 인용·주석

1.7.1 학술인용

본연구는 Smith et al. (2023) 의연구 [1] 를기반으로하며, 특히딥러닝기반문서분석에관한최신동향을반영하였다.

> **중요한발견사항**
>
> 기존연구들과달리, 본연구에서는다중모달접근법을통해텍스트와이미지정보를동시에처리하는새로운방법론을제시한다 (Johnson & Lee, 2024)[2].

1.7.2 각주참조

현재 PDF 처리시장의규모는연간 $2.3B[3] 에달하며, 향후 5 년간 CAGR 15.2% 의성장이예상된다 □.

1.7.3 참고문헌

참고문헌및주석

[1] Smith, J., Brown, M., & Davis, R. (2023). "Deep Learning Approaches for Document Analysis: A Comprehensive Survey." *Journal of AI Research*, 45(3), 123-145.

[2] Johnson, A., & Lee, K. (2024). "Multimodal Document Understanding with Transformer Networks." *Proceedings of ICCV 2024*, pp. 234-241.

[3] Global PDF Processing Market Report 2024, TechAnalysis Corp.

□ Market Research Future, "PDF Software Market Research Report - Global Forecast to 2029"

1.7.4 법적고지사항

□ 면책조항

본문서에포함된정보는일반적인정보제공목적으로만사용되며, 전문적인조언을대체하지않습니다. 본정보의사용으로인해발생하는어떠한손실이나손해에대해서도책임을지지않습니다.

1.7.5 라이선스정보

본문서는다음라이선스하에배포됩니다:

- **Creative Commons Attribution 4.0 International License**
- **MIT License** (코드부분)
- **Apache License 2.0** (데이터부분)

1.8 □ 메타데이터

속성	값
문서제목	다양한콘텐츠가혼재하는기술문서
작성자	AI Assistant
생성일	2024-06-24
버전	1.0.0
언어	한국어, 영어, 중국어, 일본어
형식	Markdown
문자수	약 15,000 자
페이지수	예상 25-30 페이지 (PDF 변환시)
키워드	다중언어, 복잡표, 수식, 프로그래밍, 차트

이문서는 *PDF* 변환테스트를위한종합적인샘플문서입니다. 실제사용시에는해당분야의전문가와상담하시기바랍니다.