多様なコンテンツが混在する技術文書

☒ 目次

☒ 数式・化学式セクション

数学的表現

基本数式　線形代数における固有値問題：$Ax = \lambda x$

ここで、A は $n \times n$ 行列、λ は固有値、x は固有ベクトルを表す。

微積分　関数 $f(x) = x^2 + 2x + 1$ の導関数：$f'(x) = 2x + 2$

定積分の計算：$\int_0^1 x^2 dx = [x^3/3]_0^1 = 1/3$

統計・確率　正規分布の確率密度関数：$f(x) = (1/\sqrt{(2\pi\sigma^2)}) \times e\text{\^{}}(-(x-\mu)^2/(2\sigma^2))$

ベイズの定理：$P(A|B) = P(B|A) \times P(A) / P(B)$

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：$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$

酸塩基反応：$HCl + NaOH \rightarrow NaCl + H_2O$

物理化学定数

定数	記号	値	単位
アボガドロ数	N☒	6.022 × 10^{23}	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/s^{2}

☒ プログラミングコード

Python 実装例

データ処理パイプライン

```python
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')
```

機械学習モデル

```
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}
```

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;
```

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;
```

☒ 多言語混在テキスト

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)

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

☒☒☒ ☒☒　☒☒ ☒☒ ☒☒☒☒ ☒☒☒ ☒☒

PDF ☒☒ ☒☒☒☒☒☒☒☒ ☒☒ ☒☒☒☒ ☒☒☒ ☒☒ ☒☒☒ ☒☒☒☒ ☒☒☒☒:

1. ☒☒☒☒ ☒☒: ☒☒☒ ☒☒ ☒☒
2. ☒ ☒☒ ☒☒: ☒☒☒ ☒ ☒☒
3. ☒☒ ☒☒ ☒☒: ☒☒☒ ☒☒ ☒☒

☒☒ ☒☒ ☒ ☒☒

☒☒ ☒☒　∀x ∈ ☒, ∃y ∈ ☒ such that x ☒ y

☒☒ ☒☒

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

☒☒ ☒☒

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

☒ ☒☒☒ ☒ ☒☒

☒☒ ☒☒ ☒☒☒

☒☒			2024 ☒ ☒☒☒☒☒(☒☒☒)				☒☒☒☒	
			Q1	Q2	Q3	Q4	☒☒☒	☒☒☒
☒☒☒	☒☒☒	☒☒	1,200	1,350	1,280	1,420	+180	+15.2%
		☒☒	2,100	2,280	2,150	2,380	+290	+14.1%
		☒☒	3,800	4,200	4,100	4,500	+650	+16.9%
		☒☒	890	920	980	1,050	+140	+15.8%
		☒☒	**7,990**	**8,750**	**8,510**	**9,350**	**+1,260**	**+15.9%**
☒☒		☒☒	5,200	5,800	5,600	6,100	+880	+16.9%
		☒☒☒	1,100	1,200	1,180	1,280	+160	+14.5%
		☒☒	**6,300**	**7,000**	**6,780**	**7,380**	**+1,040**	**+16.5%**
☒☒		☒☒	1,800	1,950	1,900	2,080	+230	+12.4%
		☒☒☒	1,200	1,300	1,250	1,380	+180	+15.0%
		☒☒	1,500	1,620	1,580	1,720	+200	+13.3%
		☒☒	800	850	820	900	+110	+13.8%
		☒☒	**5,300**	**5,720**	**5,550**	**6,080**	**+720**	**+13.6%**
			☒☒	**19,590**	**21,470**	**20,840**	**22,810**	**+3,020**

☒☒☒☒ ☒ ☒☒ ☒☒☒

⊠⊠⊠⊠	⊠⊠⊠		⊠⊠(⊠⊠⊠)		⊠⊠⊠	⊠⊠
	⊠⊠	⊠⊠	2024 ⊠	2025 ⊠		
AI ⊠⊠⊠	15 ⊠	8 ⊠	1,200	800	85%	⊠ ⊠⊠
⊠⊠⊠⊠ ⊠⊠	12 ⊠	15 ⊠	800	400	62%	⊠ ⊠⊠
⊠⊠ ⊠⊠	8 ⊠	12 ⊠	600	300	78%	⊠ ⊠⊠
⊠⊠⊠ ⊠	10 ⊠	5 ⊠	500	200	45%	⊠ ⊠⊠

⊠⊠⊠⊠ ⊠⊠⊠ ⊠⊠⊠

⊠⊠	⊠⊠	⊠⊠	⊠⊠	⊠⊠⊠	⊠⊠⊠	⊠⊠⊠⊠
⊠⊠ A	100	1,500	150,000	10%	15,000	135,000
⊠⊠ B	75	2,200	165,000	15%	24,750	140,250
⊠⊠ C	50	3,000	150,000	5%	7,500	142,500
⊠⊠ D	200	800	160,000	20%	32,000	128,000
		⊠⊠	**625,000**		**79,250**	**545,750**
				⊠⊠⊠ (10%)		**54,575**
				⊠ ⊠⊠⊠⊠		**600,325**

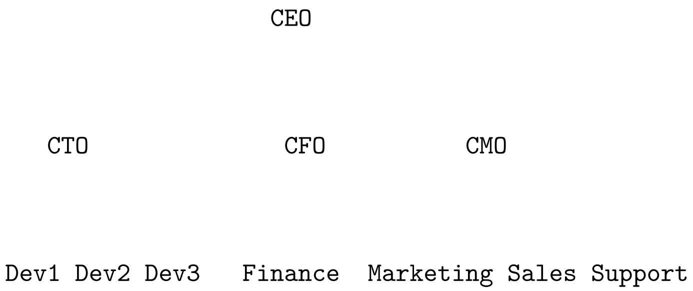
⊠ ⊠⊠·⊠⊠

ASCII ⊠⊠ ⊠⊠

⊠⊠ ⊠⊠ ⊠⊠

```
⊠⊠⊠ (⊠⊠)
    ↑
100

 90

 80

 70

 60

 50  ____________________→
    1⊠  2⊠  3⊠  4⊠  5⊠  6⊠  7⊠  8⊠  9⊠  10⊠
```

⊠⊠⊠



☒☒☒ ☒☒ (☒☒ ☒☒)

☒☒	1☒	2☒	3☒	4☒	5☒
☒☒	█				
☒☒	█	█	█		
☒☒		█	█	█	█
☒☒☒			█	█	█
☒☒				█	█

☒ ☒☒·☒☒

☒☒ ☒☒

☒ ☒☒☒ Smith et al. (2023) ☒ ☒☒ [1] ☒ ☒☒☒☒ ☒☒, ☒☒ ☒☒☒ ☒☒ ☒☒ ☒☒☒ ☒☒ ☒☒ ☒☒☒ ☒☒☒☒☒.

> ☒☒☒ ☒☒☒☒
>
> ☒☒ ☒☒☒☒ ☒☒, ☒ ☒☒☒☒☒ ☒☒ ☒☒ ☒☒☒☒ ☒☒ ☒☒ ☒☒ ☒☒☒ ☒☒☒ ☒☒☒ ☒☒☒☒ ☒☒☒ ☒☒☒☒ ☒☒☒☒ (Johnson & Lee, 2024)[2].

☒☒ ☒☒

☒☒ PDF ☒☒ ☒☒☒ ☒☒☒ ☒☒ $2.3B[3] ☒ ☒☒☒, ☒☒ 5 ☒☒ CAGR 15.2% ☒ ☒☒☒ ☒☒☒☒ ☒.

☒☒☒☒

☒☒☒☒ ☒ ☒☒

[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”

☒☒ ☒☒☒☒

☒ ☒☒ ☒☒

☒ ☒☒☒ ☒☒☒ ☒☒☒ ☒☒☒☒ ☒☒ ☒☒ ☒☒☒☒☒ ☒☒☒☒, ☒☒☒☒ ☒☒☒ ☒☒☒☒ ☒☒☒☒. ☒ ☒☒☒ ☒☒☒☒ ☒☒ ☒☒ ☒☒ ☒☒☒ ☒☒☒☒ ☒☒☒ ☒☒☒☒ ☒☒☒ ☒☒ ☒☒☒☒.

☒☒☒☒ ☒☒

☒ ☒☒☒ ☒☒ ☒☒☒☒ ☒☒ ☒☒☒☒☒:

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

☒ ☒☒☒☒☒

☒☒	☒
☒☒ ☒☒	☒☒☒ ☒☒☒☒ ☒☒☒☒ ☒☒ ☒☒
☒☒☒	AI Assistant
☒☒☒	2024-06-24
☒☒	1.0.0
☒☒	☒☒☒, ☒☒, ☒☒☒, ☒☒☒
☒☒	Markdown
☒☒ ☒	☒ 15,000 ☒
☒☒☒ ☒	☒☒ 25-30 ☒☒☒ (PDF ☒☒ ☒)
☒☒☒	☒☒☒☒, ☒☒☒, ☒☒, ☒☒☒☒☒, ☒☒

☒ ☒☒☒ PDF ☒☒ ☒☒☒☒ ☒☒ ☒☒☒☒ ☒☒ ☒☒☒☒☒. ☒☒ ☒☒ ☒☒ ☒ ☒☒ ☒☒☒ ☒☒☒☒ ☒☒☒☒☒ ☒☒☒☒.