



# ADOBE INDIA HACKATHON

CONNECTING THE DOTS...



## ROUND 1

Submission Date:  
July 28, 2025

# 「点をつなぐ」チャレンジへようこそ

## 読書を再考し、知識を再発見する

PDFを開くたびに、そこに座って-それあなたに話しかけた、関連するアイデア、そしてナレーションされた意味ライブラリ全体にわたってですか？

それが私たちが築いている未来であり、私たちが望んでいることですあなたそれを形作るのを助けるために。

の中で点をつなぐチャレンジあなたの使命は、PDFをインテリジェントでインタラクティブな体験として再創造することです。構造を理解する、洞察を明らかにする、そして応答するあなたへ信頼できる研究仲間のよう。

## これからの旅

- ラウンド1:

まずは頭脳を鍛えることから始めましょう。生のPDFから構造化されたアウトラインを驚異的なスピードと高精度で抽出します。そして、セクションを理解し、関連するアイデアを結び付けるデバイス上のインテリジェンスで、さらにパワーアップします。

- 第2ラウンド:

いよいよ本番です！AdobeのPDF埋め込みAPIを使って、美しく直感的な読書用ウェブアプリを構築しましょう。第1ラウンドで作成した作品を使って、未来的なウェブアプリをデザインします。

## これがなぜ重要なのか

文書が溢れる世界で勝利するのは、より多くのコンテンツではなく、コンテキスト単にツールを作るのではなく、私たちがどのように未来を築くかという未来を創っているのです。読んで、学び、つながるMLハッカー、UIビルダー、インサイトウィスパラーなど、どのようなバックグラウンドであっても、ここがあなたのステージです。

## 参加しますか？

行間を読む時が来ました。点と点を繋ぎ、まるで魔法。さあ行こう。

# ラウンド1A: ドキュメントを理解する

チャレンジテーマ: ドキュメントを通して点と点をつなぐ

## あなたの使命

PDF文書を渡されますが、ただ読むのではなく、機械のように理解することが求められます。あなたの仕事は、文書の構造化されたアウトライン、つまり**タイトル**、そして次のような見出し**H1**、**水素**、そして**H3**—クリーンで階層的な形式で。

この概要は**財団**ハッカソンの残りの旅のために。

## これがなぜ重要なのか

PDFはどこにでも存在しますが、機械はPDFの構造を自然に理解することはできません。アウトライン抽出ツールを構築することで、セマンティック検索、レコメンデーションシステム、インサイト生成など、よりスマートなドキュメントエクスペリエンスを実現できます。

## 構築に必要なもの

次のようなソリューションを構築する必要があります。

- 受け入れる**PDFファイル**（最大50ページ）
- 抜粋:
  - **タイトル**
  - **見出し**: H1、H2、H3（レベルとページ番号付き）
- 以下の形式で有効な JSON ファイルを出力します。

```
{  
  「タイトル」: 「AIを理解する」、 「アウ  
  トライン」: 「  
    { "level": "H1", "text": "はじめに", "page": 1 },  
    { "level": "H2", "text": "AIとは何か", "page": 2 },  
    { "level": "H3", "text": "AIの歴史", "page": 3 }  
  ]  
}
```

## あなたには提供される

1. Aサンプル入力PDF（例：sample.pdf）
2. Aサンプルのグラウンドトゥールズ出力（フォーマットを明確にするためにsample.jsonを使用します）

### 3. サンプルDockerfile

#### 4. サンプルソリューション

## Dockerの要件

- DockerfileがAMD64アーキテクチャに対応していることを確認してください。イメージはAMD64マシン上でビルド・実行されるため、ベースイメージと依存関係はlinux/amd64をサポートしている必要があります。必要に応じて、Dockerfileに以下の記述を追加してプラットフォームを明示的に指定することもできます：FROM -- platform=linux/amd64 <base\_image>
- **CPUアーキテクチャ**:amd64 (x86\_64)
- GPU依存なし
- モデルサイズ（使用する場合） ≤ 200MB
- オフラインで動作し、ネットワーク/インターネット通話は不可

## 予想される実行

次のコマンドを使用して Docker イメージをビルドします。

```
`` docker build --platform linux/amd64 -t  
mysolutionname:somerandomidentifier ``
```

イメージをビルドした後、送信された手順で指定された実行コマンドを使用してソリューションを実行します。

```
`` docker run --rm -v $(pwd)/input:/app/input -v $(pwd)/output:/app/output --  
network none mysolutionname:somerandomidentifier ``
```

コンテナは次のようになります。

- /app/input ディレクトリからすべての PDF を自動的に処理し、各 filename.pdf に対応する filename.json を /app/output に生成します。
- 出力.json

## 制約

制約	要件
実行 時間	50ページのPDFの場合は10秒 以内

モデルサイズ	≤ 200MB（使用する場合） イ
ネットワーク	インターネットアクセスなし
	許可された
ランタイム	CPU (amd64) 上で実行する必要があります。 ソリューションは、8 つの CPU と 16 GB の RAM 構成のシステム で実行する必要があります。

## 採点基準

基準	マックス ポイント
方位検出精度（精度 + 再現率）	25
パフォーマンス（時間とサイズのコンプライアンス）	10
ボーナス：多言語対応（例：日本語）	10
合計	45

## 提出チェックリスト

1. ルートディレクトリに動作するDockerfileがあるGitプロジェクトと
2. 作業中の**Dockerファイル**
3. コンテナ内にインストールされたすべての依存関係
4. 次の内容を説明する README.md:
  - o あなたのアプローチ
  - o 使用されたモデルまたはライブラリ
  - o ソリューションを構築して実行する方法（これは純粋にドキュメント作成の目的上、ソリューションは上記の「想定される実行」セクションを使用して実行する必要があります。

### プロのヒント

- 見出しレベルの決定にはフォント サイズだけに頼らないでください。一部の PDF の見出しではその前提が崩れます。
- シンプルな PDF と複雑な PDF の両方でソリューションをテストします。
- コードをモジュール化します。この構造はラウンド 1B で再利用されます。

- 重要 – コンテストの締め切りまで Git リポジトリを非公開にしておいてください。リポジトリを公開するタイミングについては通知されます。

#### してはいけないこと

- 見出しやファイル固有のロジックをハードコードしないでください
- APIやウェブ呼び出しを行わない
- ランタイム/モデルサイズの制約を超えないようにする

[[パブリックデータセットフォルダ]]

(サンプル入力ファイルと出力ファイルについては、付録を参照してください)

## ラウンド1B: ペルソナ主導のドキュメントインテリジェンス

テーマ：「大切なものをつなぐ — 大切なユーザーのために」

### チャレンジ概要（参加者向け）

特定のペルソナとその実行すべき仕事に基づいて、ドキュメントのコレクションから最も関連性の高いセクションを抽出し、優先順位を付ける、インテリジェントなドキュメントアナリストとして機能するシステムを構築します。

### 入力仕様

**2.文書コレクション:** 3～10件の関連PDFペルソナの定義: 特定の専門知識と重点分野を含む役割の説明

**3.完了すべき仕事:** ペルソナが達成する必要がある具体的なタスク

ドキュメントコレクション、ペルソナ、そしてJTOB（ジョブ・トゥ・ビー・ダン）は非常に多様です。そのため、チームが構築する必要があるソリューションは、この多様性に対応できるよう汎用性が高くなければなりません。

- ・ 文書はあらゆる分野のものが可能です (例: 研究論文、学校/大学の書籍、財務レポート、ニュース記事など)
- ・ ペルソナも非常に多様です (例: 研究者、学生、営業担当者、ジャーナリスト、起業家など)
- ・ 実行すべき仕事: これはペルソナに関連します (例: 特定のトピックと入手可能な研究論文に関する文献レビューを提供する、化学文書に基づいて有機化学について何を勉強すべきか、詳細な年度末財務報告書に基づいて企業 XYZ の財務を要約するなど)。

### サンプルテストケース

#### テストケース1: 学術研究

- ・ **文書:** 「創薬のためのグラフニューラルネットワーク」に関する4つの研究論文
- ・ **ペルソナ:** 計算生物学博士研究員
- ・ **仕事:** 「方法論、データセット、パフォーマンスベンチマークに焦点を当てた包括的な文献レビューを準備する」

## テストケース2: ビジネス分析

- ・ **文書:** 競合するテクノロジー企業からの3つの年次レポート（2022～2024年）
- ・ **ペルソナ:** 投資アナリスト
- ・ **仕事:** 「収益動向、研究開発投資、市場ポジショニング戦略を分析する」

## テストケース3: 教育コンテンツ

- ・ **文書:** 有機化学の教科書から5章
- ・ **ペルソナ:** 学部化学専攻の学生
- ・ **仕事:** 「反応速度論の試験準備のための重要な概念とメカニズムを特定する」

## 必要な出力

- ・ 出力JSON形式: 参照 [チャレンジ1b\\_出力.json](#)

出力には次の内容が含まれます。

1. メタデータ:
  - a. 入力文書
  - b. ペルソナ
  - c. やるべき仕事
  - d. 処理タイムスタンプ
2. 抜粋:
  - a. 文書
  - b. ページ番号
  - c. セクションタイトル
  - d. 重要度ランク
3. サブセクション分析:
  - a. 文書
  - b.
  - c. 洗練されたテキスト
  - d. ページ番号制約

- ・ 実行する必要がありますCPU上のみ
- ・ モデルサイズ ≤ 1GB
- ・ 処理時間 ≤ 60秒書類収集用（3～5件）
- ・ インターネットアクセスなし実行中に許可される



## 成果物

- approach\_explanation.md（方法論を説明する300～500語）
- Dockerfileと実行手順

テスト用のサンプル入力/出力

•

## 採点基準

基準	マックス ポイント	説明
セクションの関連性	60	選択されたセクションが適切なスタック ランキングでペルソナと職務要件にどの 程度一致しているか
サブセクションの関連性	40	細分化されたサブセクションの抽 出とランキングの品質

付録：

<https://github.com/jhaaj08/Adobe-India-Hackathon25.git>