Final Reflection

During the development of this project, I encountered several challenges when working with different Azure Cognitive Services. Two specific resources posed significant issues, and I describe them below:

1. Face API – Regional Limitations and Access Restrictions

While implementing face verification using the Azure Face API, I attempted to use the verify_face_to_face functionality to compare two face images. Despite correctly setting up the endpoint and authentication, the API consistently returned the error:

"UnsupportedFeature: Feature is not supported, missing approval for one or more of the following features: Identification,Verification. Please apply for access at https://aka.ms/facerecognition"

After investigating the issue, I discovered that **Face Identification and Verification are restricted in the European region** due to privacy and compliance regulations. Azure requires explicit approval for these features, and a justification must be submitted through the link provided in the error message.

Due to time constraints, I was unable to complete the approval process during the project timeline. This limitation significantly impacted on the ability to fully demonstrate the person-matching feature using Azure Face API.

2. Document Intelligence – Handling File Format and SAS Access Errors

For the document analysis component, I used **Azure Document Intelligence** (formerly Form Recognizer) to extract information from ID cards and boarding passes.

One of the most common problems was related to the document input itself. When using begin_analyze_document_from_url, I initially attempted to submit a **PDF hosted on Azure Blob Storage**, but the API responded with:

```
InvalidImage: Decoding error, image format unsupported.
```

The root cause was that the **prebuilt models for ID recognition do not support PDFs**, only images (e.g., JPG or PNG). Additionally, improperly constructed **SAS URLs (Shared Access Signatures)** led to repeated 400/403 HTTP errors.

To resolve this:

- I verified the content type with requests.head() before submitting.
- I ensured the SAS token included the correct reading permissions (sp=r) and had not expired.
- I switched from PDF to PNG files to compatibility with the prebuilt ID model.

This experience reinforced the importance of **matching input formats with model expectations** and using proper diagnostics when working with Azure Cognitive Services.