**Script Oral Présentation PDF Alteryx :**

**Slide 0 :**

Bonjour, je suis Lila Roig, Alternante en Data Science à la Caisse d’Epargne et je vais vous présenter brièvement 2 projets de Data Science qui ont été réalisés.

Le premier projet consiste à implémenter une méthode pour détecter la présence de signatures sur plusieurs types de documents PDF.

**Slide 1 :**

Clic 1 : Nous voulons détecter la présence d’une signature sur plusieurs types de PDF :

* le document LEA (synthèse de notre proposition d'épargne),
* EAI (document d'auto-certification),
* le BS (bulletin de souscription) et d’autres documents.
* …

Pour cela, nous devons **considérer plusieurs éléments**:

Clic 2 :

* *Il existe plusieurs versions possibles pour un type de document donné* -> par exemple, il existe plusieurs **versions** du document EAI.
* *Les documents peuvent ne pas correspondre au type indiqué* -> un document peut être marqué comme étant EAI alors que c’est un autre document
* *il faut que notre méthode puisse identifier* ***quel type de document*** *que l’on a*

Clic 3 :

* *Un document PDF peut contenir plusieurs pages*
* *Et la signature ne se trouve pas toujours sur la dernière page du document*
* *Il faut que l’on puisse identifier quelle est la page contenant la signature pour savoir quelle page on va* ***analyser****.*

Clic 4 :

* Enfin, certains documents sont mal scannées (rotation, rognées)
* *Il faut pouvoir traiter ces cas*

Clic 5 : Pour ce **projet**, nous avons utilisé 2 approches différentes.

* Un premier code en python
* Une 2e approche avec Alteryx et le module supplémentaire Intelligence Suite, que je **vais détailler ici**.

Clic 6 : Les 2 méthodes se basent sur des **OCR** qui sont des **réseaux de neurones** permettant de **traduire des images en texte**.

Clic 7 : Pour le moment, nous pu tester nos méthodes sur les documents LEA et EAI

**Slide 2** :

Voici le Workflow Alteryx pour le document EAI dans son ensemble dont je vais présenter chaque étape.

**Slide 3** :

Pour ce projet, nous utilisons le module **Computer Vision** d’Intelligence Suite qui **possède les outils affichés ici**.

Clic 1 : **La** **première étape** consiste à charger les PDF avec l’outil Image Input puis à affectuer un traitement d’image avec l’outil Image Processing, ce qui permet de faciliter la lecture du texte par l’OCR.

Clic 2 : **A la 2ème étape**, nous **souhaitons** identifier la page contenant la signature.

**Pour cela**, on utilise **d’abord** l’outil Image to Text pour lire le texte des PDF.

Clic 3 : **Puis**, on recherche **dans le texte extrait**, la présence de **certains** mots clés, **spécifiques au document (ici le document EAI)**.

Clic 4 : Si la page **que l’on analyse** possède suffisamment de mots clés alors il s’agit de la page contenant la signature et on poursuit l’analyse,

Sinon

si **une fois toutes les pages analysées**, on a pas trouvé **suffisamment de mots clés** , **on considère** que le PDF est illisible ou que la page supposées contenir la signature **n’existe pas** et donc **il s’agit peut-être du mauvais document.**

**Slide 4** :

On crée maintenant les modèles pour le document EAI. **Comme le document EAI possède 2 versions différentes, on doit créer 2 modèles**.

Clic 1 : On utilise l’outil Image Template pour dessiner des zones de vérification sur chaque modèle.

Clic 2 : et les rectangles supposés contenir la signature.

Clic 3 : Pour savoir si le PDF **n’est pas décalé** ou s’il s’agit du bon document**:**

On regarde le **texte** contenu dans les zones violettes.

1. Si les **mots détectés** dans les **zones** **violettes** du **modèle 1** sont **corrects**, la page **qu’on analyse est bien identique au modèle 1** -> on poursuit l’analyse

Sinon

1. on est peut-être en **présence du modèle 2**? Si les **mots détectés** dans les **zones** **violettes** du **modèle 2** sont **corrects,** il s’agit bien du **modèle 2** -> on poursuit l’analyse

Sinon

cela veut dire que le PDF est décalé ou alors le PDF est différent des 2 modèles.

**Slide 5** :

**Enfin, on regarde le texte de zones vertes**.

Clic 1 : Si ce texte est suffisamment long, le document est considéré comme signé.

Sinon

on utilise l’outil Image Profile pour calculer le remplissage des zones vertes.

Clic 2 : Si ce remplissage est suffisant, le PDF est signé, sinon, il n’est pas signé.

Clic 3 :

Nous avons effectué des premiers tests avec python et Alteryx sur 1000 LEA.

Nous n’avons **pas pu vérifier toutes les prédictions**, donc nous avons ouvert uniquement les PDF qui présentaient une **anomalie**.

* Le workflow alteyrx tourne en 16h40 tandis que le code python tourne en 2h30.
* Les 2 méthodes ont bien trouvé les 3 documents non signés
* et les **3 cas où le document n’était** **pas le bon** (PDF illisible pour Alteryx et page non trouvée pour Python).
* Alteryx effectue 51 erreurs sur les documents vérifiés. Les erreurs sont essentiellement pour les cas où les PDF sont décalés.
* Python effectue 2 erreurs sur les documents vérifiés : 2 cas où Il n’arrive pas bien à trouver la page du PDF.

**Python ne présente pas d’erreurs de PDF décalés car la principale différence entre les 2 méthodes est que dans le code Python, le cadre s’adapte à chaque PDF tandis que dans Alteyrx, les cadres sont fixes.**

**Slide 6** :

Nous avons ensuite effectué des tests sur 100 documents EAI et 100 documents LEA où nous avons pu ici **vérifier toutes les prédictions**.

Pour le **EAI**, nous avons

* Pour Alteyrx 1h de temps de calcul et 3 erreurs dont 1 erreur « grave » lorsque l’on prédit que le document est signé alors qu’il ne l’est pas.
* Et pour Python 17 min de temps de calcul et 16 erreurs.

Les erreurs de Python sont surtout dues lors du traitement des documents numériques où il dit que le document est non signé alors qu’il l’est.

Pour le **LEA**, nous avons

* Pour Alteryx 43min de temps de calcul et 5 erreurs
* Et pour Python 18 min de temps de calcul et 11 erreurs. Les erreurs encore une fois lorsque le code dit que le document n’est pas signé alors qu’il l’est.

**Pour python, les erreurs peuvent être résolues en modifiant la tolérance** lors du calcul du remplissage des cadres supposés contenir la signature.

On peut aussi implémenter une méthode pour **détecter en amont les documents numériques.**

**Slide 7** :

Dans cette slide, on présente les avantages et inconvénients de chaque méthode.

Rapidité d’adaptation à un nouveau document : On a passé du temps à développer la méthode pour le premier document mais maintenant adapter la méthode à de nouveau documents est rapide.

**Script Oral Présentation MotsInterdits :**

**Slide 0 :**

Je vais maintenant parler rapidement du 2ème projet où l’objectif est de détecter les mots interdits dans le texte de la table Osirisk.

**Slide 1** :

Clic 1 :

* Nous avons recherché la présence des mots interdits dans 6 colonnes la table Osirisk contenant 50008 lignes
* Pour rechercher les mots interdits, nous avons utilisé les deux listes de mots interdits fournis par la DPO.

Clic 2-4 :

Pour la première méthode utilisée,

* nous avons tout d'abord effectué un pré-traitement des mots interdits et du texte de la table Osirisk (retrait des accents, de la ponctuation, des chiffres, des valeurs manquantes, des majuscules).
* Puis nous avons recherché les mots interdits présents dans la table Osirisk.
* Cependant, certains noms tels que 'Nègre Véronique' sont considérés à tort comme des mots interdits. Ainsi, lorsque nous recherchons les mots interdits qui sont dans Osirisk, nous considérons qu'un mot n'est pas un mot interdit si celui-ci est **précédé ou suivi d'un prénom**.

Clic 3-6:

* La deuxième méthode utilisée s'apparente à la première méthode mais inclut de plus une étape supplémentaire qui consiste à lemmatiser (prendre la forme neutre canonique du mot) les mots interdits et la table Osirisk.

**Ceci permet de capturer davantage de mots interdits.**

* Puis on réaliser les mêmes étapes que précédemment

Clic 7 :

Avec la première méthode, nous avons trouvé 372 mots interdits et 441 avec la deuxième méthode.

Clic 8 :

Les 2 première méthodes ne permettent pas de traiter les **cas nécessitant une compréhension du texte**, comme pour cette phrase : … où blaireau est considéré à tort comme un mot interdit

On pourrait implémenter une 3ème méthode en utilisant des **embeddings** **et/ou des réseaux de neurones** ayant une **meilleure compréhension du texte**.