

Explication Détaillée de l'Application Streamlit : Analyse de Clients et Fournisseurs

Cette application est une solution complète d'analyse de données financières, axée sur les transactions par chèque des clients et des fournisseurs. Elle combine des techniques d'analyse exploratoire de données (EDA), d'apprentissage automatique (Machine Learning) pour la détection d'anomalies et la segmentation, et l'intelligence artificielle générative (LLM) pour une interaction conversationnelle avec les données.

I. Vue d'Ensemble des Fonctionnalités (Features)

L'application est structurée en plusieurs sections, offrant une expérience utilisateur riche et ciblée :

- Aperçu des Données Brutes et Préparées
- Exploration Détaillée des Données Clients/Fournisseurs
- Analyse Approfondie des Paiements Fractionnés
- Détection d'Anomalies (IA)
- Clustering et Segmentation (IA)
- Analyse de la Relation Clients-Fournisseurs
- Discuter avec les Données (LLM)
- Télécharger des Exemples de Questions LLM

II. Détails Techniques Approfondis

1. Pré-traitement et Ingénierie des Caractéristiques

Chargement des Données

- Source : Client - Sheet1.csv, Fournisseur - Sheet1.csv
- Utilisation de `@st.cache_data` pour optimiser les performances

Nettoyage et Conversion

- Colonnes de dates transformées avec gestion d'erreurs (`errors='coerce'`)
- Colonnes vides supprimées : `Somme_Impaye`, `Nombre_Impaye`

Gestion des Valeurs Manquantes

- Catégorielles : Remplacées par 'Inconnu'
- Numériques : Imputation par la médiane (`Salaire`, `Montant_cheque`, `Total montant cheque`, `Nombre`)

Création de la Caractéristique `is_split_record`

- Binaire : 1 si `Nombre > 1`, sinon 0

2. Agrégation des Caractéristiques par Entité (`aggregate_features`)

L'objectif est de passer du niveau transaction à un niveau entité (client/fournisseur).

Identifiants

- Client : `Compte_key_Payeur`
- Fournisseur : `Compte_Key`

Variables Utilisées dans l'Agrégation

- `Montant_cheque`
- `Nombre`
- `Date_Operation`
- `is_split_record`

Caractéristiques Agrégées

Nom de la Variable	Description
<code>total_montant</code>	Somme des <code>Montant_cheque</code>
<code>moy_montant_cheque</code>	Moyenne de <code>Montant_cheque</code>
<code>max_montant_cheque</code>	Maximum de <code>Montant_cheque</code>
<code>min_montant_cheque</code>	Minimum de <code>Montant_cheque</code>
<code>freq_transactions</code>	Somme de <code>Nombre</code> (fréquence totale des chèques)
<code>nb_operations_distinctes</code>	Nombre de dates d'opération uniques (<code>Date_Operation</code>)
<code>recence</code>	Nombre de jours depuis la dernière opération
<code>nb_paiements_fractionnes_records</code>	Somme des <code>is_split_record</code>
<code>score_fragmentation</code>	$\text{freq_transactions} / \text{nb_operations_distinctes}$ (avec division par zéro gérée)

3. Visualisation des Données (EDA)

- **matplotlib, seaborn**
 - Histogrammes, count plots, line plots avec groupement par `Annee_Mois`
-

4. Intelligence Artificielle et Machine Learning

PCA

- Réduction à 2 composantes (`pca1, pca2`)
- `StandardScaler` pour normalisation

Isolation Forest (Détection d'Anomalies)

- **Modèle** : `sklearn.ensemble.IsolationForest`
- **Paramètre clé** : `contamination`
- **Visualisation** : anomalies en rouge sur graphes PCA et scatter plots

K-Means (Segmentation)

- **Modèle** : `sklearn.cluster.KMeans`
 - Choix du nombre de clusters via `slider`
 - Affichage des centroïdes (valeurs inversées post-scaling)
-

5. Analyse de la Relation Clients-Fournisseurs

- Recherche des entités présentes dans les deux fichiers
 - Corrélations comportementales (`sns.heatmap`)
 - Agrégation double vue (client + fournisseur)
-

6. Intégration LLM (Gemini 2.0 Flash)

- **Modèle utilisé** : `google.generativeai.GenerativeModel("gemini-2.0-flash")`
- Prompt système riche : rôle, description des datasets, `df.describe()` + `df.info()`
- Conversation via boîte de chat avec gestion d'historique (`st.session_state.chat_history`)
- Gestion des erreurs avec `try/except`