Département d'informatique et de mathématique
8PRO408 - Outils de programmation pour la science des données
Chargé de Cours : HN Doukaga, hndoukag@uqac.ca

Mini-Projet : Analyse exploratoire d'un jeu de données bancaire

Analyse exploratoire d'un jeu de données réel : transactions bancaires et détection de fraude

Description générale

Vous utiliserez un jeu de données réel contenant 284 807 transactions bancaires (dont 492 fraudeuses) afin de réaliser une **analyse exploratoire complète**. L'objectif est de comprendre la structure du dataset, d'identifier les tendances principales, d'analyser le déséquilibre des classes et de produire des visualisations pertinentes.

Aucun modèle de classification n'est demandé : il s'agit **uniquement d'un travail d'EDA structuré**, conforme aux bonnes pratiques vues en cours.

Objectifs du mini-projet

- Importer, inspecter et documenter un dataset réel.
- Réaliser un nettoyage de base si nécessaire.
- Analyser la distribution des variables (Time, Amount, V1–V28).
- Visualiser les relations entre variables, incluant le caractère fortement déséquilibré de la classe *fraude*.
- Présenter les résultats dans un notebook clair et reproductible.

Travail demandé

Vous devez :

1. **Charger et explorer le dataset**
   - Aperçu général : colonnes, types, valeurs manquantes, statistiques descriptives.
2. **Analyser la distribution des variables**
   - Distributions simples (histogrammes, boxplots).
   - Analyse temporelle de la variable *Time*.
   - Analyse du montant des transactions.
3. **Examiner la variable cible (Class)**
   - Répartition des classes.
   - Visualisation du déséquilibre.
4. **Explorer les relations entre variables**
   - Corrélations globales.
   - Visualisations ciblées pour mieux comprendre la structure du dataset.
5. **Produire des visualisations variées**
   - Obligatoire : Matplotlib + Seaborn + Plotly.
6. **Synthétiser vos conclusions**
   - Ce que vous avez observé.

- Les caractéristiques des transactions frauduleuses (montants, patterns, etc.).
- Les limites du dataset (PCA, confidentialité, déséquilibre).

Livrables

À déposer sur GitHub (obligatoire) :

- Un **notebook Jupyter** (.ipynb) contenant l'ensemble de l'analyse.
- Un **rapport PDF court (max 2 pages)** résumant vos observations principales.
- Une **courte application Streamlit** affichant 2 à 4 visualisations interactives.
- Un fichier README.md expliquant comment exécuter le projet.

Modalités de réalisation

- Travail **individuel** ou en **équipes de 2 à 4**.
- Date limite : **10 décembre 2025 à 23h59**.
- Le dépôt GitHub doit être public ou accessible via un lien.

Critères d'évaluation (100 points)

- Clarté du notebook et qualité des analyses. **/25**
- Qualité et variété des visualisations. **/15**
- Pertinence des observations. **/25**
- Propreté et organisation du dépôt GitHub. **/15**
- Fonctionnalité minimale de l'application Streamlit. **/10**
- Qualité du rapport PDF (synthèse, structure, précision). **/10**

Pénalités

- Mauvais français : −20 %
- Retard : −10 %, puis −5 % par jour (max 7 jours)
- Non-respect des consignes GitHub : −5 à −15 %

Lien du jeu de données : Kaggle

Département d'informatique et de mathématique
8PRO408 - Outils de programmation pour la science des données
Chargé de Cours : HN Doukaga, hndoukag@uqac.ca

Mini-Projet : Analyse des contenus Netflix

Analyse exploratoire du catalogue Netflix (Movies & TV Shows)

Description générale

Vous utiliserez un jeu de données réel provenant de Netflix (8 807 titres, 12 colonnes) afin de réaliser une **analyse exploratoire structurée** du catalogue : films, séries, pays d'origine, genres, dates d'ajout, casting, etc.

L'objectif du mini-projet est de comprendre la composition du catalogue Netflix, d'explorer sa diversité et d'identifier des tendances clés (types de contenus, répartition géographique, évolution temporelle, etc.).

Objectifs du projet

- Explorer la structure et la qualité du dataset.
- Analyser les types de contenus proposés (films vs séries).
- Étudier la distribution des genres, pays, années de sortie.
- Identifier des tendances temporelles (contenus récents, ajouts par année, etc.).
- Produire des visualisations pertinentes à l'aide de Pandas, Seaborn, Matplotlib et Plotly.
- Résumer les observations sous forme d'un court rapport analytique.

Travail demandé

1. **Exploration du dataset**
   - Aperçu général : colonnes, valeurs manquantes, duplicats, types de données.
   - Vérification et nettoyage minimal (si nécessaire).
2. **Analyse des contenus**
   - Films vs séries : proportions, tendances par année.
   - Genres principaux (listed_in) : regroupement, fréquences.
   - Répartition géographique (country).
   - Casting & réalisateurs : analyse simple (comptages, noms fréquents).
3. **Analyse temporelle**
   - Distribution par année de sortie (release_year).
   - Analyse de la colonne date_added.
4. **Visualisations**
   - Représentations obligatoires :
     - histogrammes, countplots, boxplots (Seaborn / Matplotlib)
     - visualisations interactives (Plotly)
   - Choix libres : nuages de mots, diagrammes circulaires, timelines, etc.
5. **Synthèse**
   - Résumé des observations principales.

    - Tendances remarquables (contenus récents, pays dominants, genres populaires).

Livrables

À déposer sur GitHub :

- Un **notebook Jupyter** complet avec analyses et graphiques.
- Un **rapport PDF court** (1 à 2 pages).
- Une **mini application Streamlit** (visualisations interactives).
- Un fichier README.md expliquant comment exécuter le projet.

Modalités

- Travail individuel ou en équipes de 2 à 4.
- Date limite : **10 décembre 2025 – 23h59**.
- Le dépôt GitHub doit être accessible.

Évaluation (100 points)

- Qualité de l'analyse exploratoire **/25**
- Variété et pertinence des visualisations **/15**
- Clarté du notebook **/25**
- Qualité du rapport **/15**
- Fonctionnalité de l'application Streamlit **/10**
- Organisation du dépôt GitHub **/10**

Pénalités

- Mauvais français : −20 %
- Retard : −10 %, puis −5 % par jour (max 7 jours)
- Non-respect des consignes GitHub : −5 à −15 %

Liens du jeu de données : Kaggle

Département d'informatique et de mathématique
8PRO408 - Outils de programmation pour la science des données
Chargé de Cours : HN Doukaga, hndoukag@uqac.ca

Mini-Projet : Analyse temporelle du Bitcoin

Analyse exploratoire et temporelle des données historiques du Bitcoin (1-min OHLCV)

Description générale

Vous utiliserez un jeu de données réel contenant des **données historiques du Bitcoin à la minute** (plus de 7,3 millions de lignes, 6 colonnes : Timestamp, Open, High, Low, Close, Volume).
Les données couvrent la période de **janvier 2012 à aujourd'hui** (UTC, selon la version téléchargée).

L'objectif du mini-projet est de réaliser une **analyse exploratoire et temporelle structurée** : comprendre l'évolution du prix du Bitcoin, analyser la volatilité, la liquidité (volume), et préparer le terrain pour de futurs modèles de prévision.

Objectifs du projet

- Charger et manipuler un **jeu de données volumineux** (time series).
- Inspecter la qualité des données (trous, doublons, anomalies).
- Réaliser une **analyse temporelle** (par jour, mois, année, etc.).
- Étudier la dynamique du prix (Open/High/Low/Close) et du volume.
- Produire des visualisations adaptées aux séries temporelles (courbes, zooms, agrégations).
- Résumer les observations dans un court rapport.

Travail demandé

1. **Préparation et exploration initiale**
   - Aperçu du dataset : info(), describe(), nombre de lignes, types.
   - Conversion de Timestamp en date/heure lisible.
   - Vérification des valeurs manquantes, doublons, sauts de temps.
2. **Agrégations temporelles**
   - Construction de séries agrégées :
     - par heure ;
     - par jour ;
     - par mois.
   - Calcul de statistiques simples : prix moyen, min, max, volatilité simple, volume total.
3. **Analyse des tendances et volatilité**
   - Visualisation de l'évolution du prix dans le temps (courbe Close).
   - Zoom sur certaines périodes (ex. bull run, crash).
   - Analyse de la volatilité (écart-type glissant, variation relative).
   - Relation simple entre prix et volume (corrélations visuelles ou numériques).

4. **Visualisations**
    - Graphiques obligatoires :
        - séries temporelles (Matplotlib/Seaborn) ;
        - heatmap simple de corrélation ;
        - visualisations interactives (Plotly) pour zoomer dans le temps.
5. **Synthèse**
    - Principales tendances observées (hausse globale, phases de forte volatilité, périodes de volume élevé).
    - Discussion sur la qualité et les limites des données (trous, artefacts, granularité).

Livrables

À déposer sur GitHub :

- Un **notebook Jupyter** complet avec tout le code, les analyses et les graphiques.
- Un **rapport PDF court** (1 à 2 pages) résumant vos résultats.
- Une **mini application Streamlit** permettant d'explorer :
    - une courbe de prix interactive ;
    - au moins un graphique volume/prix avec filtres temporels.
- Un fichier README.md expliquant comment installer et lancer le projet.

Modalités

- Travail individuel ou en équipes de **2 à 4 étudiant·es**.
- Date limite : **10 décembre 2025 – 23h59**.
- Le dépôt GitHub doit être public ou accessible via un lien.

Évaluation (100 points)

- Qualité de l'analyse exploratoire et temporelle. **/25**
- Pertinence et lisibilité des visualisations. **/15**
- Clarté et organisation du notebook. **/25**
- Qualité du rapport PDF (synthèse, structure). **/15**
- Fonctionnalité minimale de l'application Streamlit. **/10**
- Propreté et structure du dépôt GitHub. **/10**

Pénalités

- Mauvais français : −20 %
- Retard : −10 %, puis −5 % par jour (max 7 jours)
- Non-respect des consignes GitHub : −5 à −15 %

Lien du jeu de données : Kaggle

Département d'informatique et de mathématique
8PRO408 - Outils de programmation pour la science des données
Chargé de Cours : HN Doukaga, hndoukag@uqac.ca

Mini-Projet : Analyse exploratoire de la demande hôtelière

Analyse des réservations hôtelières : City Hotel vs Resort Hotel

Description générale

Vous utiliserez un jeu de données réel contenant **119 390 réservations** (32 variables) issues d'un hôtel de ville (*City Hotel*) et d'un hôtel de villégiature (*Resort Hotel*). Le dataset comprend des informations sur les dates de réservation, durées de séjour, types de clients, prix (ADR), annulations, demandes spéciales, etc.

L'objectif du mini-projet est de réaliser une **analyse exploratoire complète** permettant de comprendre les facteurs clés influençant les réservations, les annulations et la demande générale.

Objectifs du projet

- Explorer la structure et la qualité des données.
- Comparer les deux types d'hôtels sur plusieurs indicateurs.
- Étudier les comportements de réservation (saisonnalité, durée, prix, annulation).
- Identifier les profils de clients et leurs comportements.
- Produire des visualisations pertinentes (statistiques et interactives).
- Résumer les observations dans un court rapport analytique.

Travail demandé

1. **Exploration du dataset**
   - Aperçu des colonnes, types, valeurs manquantes, duplicats.
   - Nettoyage minimal si nécessaire (dates, valeurs NaN).
2. **Comparaison City Hotel vs Resort Hotel**
   - Taux d'annulation.
   - Prix moyen (ADR).
   - Durée des séjours (weekend/weeknights).
   - Répartition des types de clients.
3. **Analyse temporelle**
   - Saisonnalité (mois, semaines).
   - Années 2015–2017 : tendances de la demande.
   - Lead time : délais entre réservation et arrivée.
4. **Analyse des comportements clients**
   - Nombre d'adultes/enfants/bébés.
   - Demandes spéciales.
   - Dépôts (deposit_type).
   - Agents / entreprises (optionnel).
5. **Visualisations obligatoires**
   - Histogrammes / countplots / boxplots (Seaborn/Matplotlib).
   - Visualisations interactives (Plotly).

    - Au moins un graphique comparatif entre les deux hôtels.
6. **Synthèse**
    - Résultats et tendances clés (annulations, ADR, saisons, types de clients).
    - Limites des données et pistes pour modélisation future.

Livrables

À déposer sur GitHub :

- Un **notebook Jupyter** complet et propre.
- Un **rapport PDF court** (1–2 pages).
- Une **mini application Streamlit** affichant quelques visualisations interactives.
- Un README.md expliquant l'exécution du projet.

Modalités

- Travail individuel ou en équipes de **2 à 4 étudiant·es**.
- Date limite : **10 décembre 2025 – 23h59**.
- Le dépôt GitHub doit être accessible.

Évaluation (100 points)

- Qualité de l'analyse exploratoire. **/25**
- Pertinence et clarté des visualisations. **/15**
- Organisation et lisibilité du notebook. **/25**
- Qualité du rapport PDF (synthèse et structure). **/15**
- Fonctionnalité de l'application Streamlit. **/10**
- Structure du dépôt GitHub. **/10**

Pénalités

- Mauvais français : −20 %
- Retard : −10 %, puis −5 % par jour (max 7 jours)
- Non-respect des consignes GitHub : −5 à −15 %

Lien du jeu de données : Kaggle

Département d'informatique et de mathématique
8PRO408 - Outils de programmation pour la science des données
Chargé de Cours : HN Doukaga, hndoukag@uqac.ca

Mini-Projet : Analyse exploratoire d'un dataset hospitalier synthétique

Analyse exploratoire des admissions hospitalières et des diagnostics médicaux

Description générale

Vous travaillerez sur un dataset **synthetique de 55 500 dossiers patients** contenant des informations d'admission, diagnostics, médecins, assurances, coûts, traitements et résultats de tests médicaux.
Aucun renseignement réel n'est présent : il s'agit d'un dataset généré pour l'analyse et l'apprentissage.

L'objectif du mini-projet est de réaliser une **analyse exploratoire complète** afin de comprendre les tendances médicales, identifier les facteurs liés aux admissions et analyser la répartition des résultats de tests médicaux.

Objectifs du projet

- Explorer la structure et la qualité des données.
- Étudier les caractéristiques des patients (âge, genre, groupe sanguin).
- Analyser les pathologies et leur fréquence.
- Étudier les metrics hospitalières : durée de séjour, coûts, types d'admission.
- Visualiser les relations entre variables médicales et administratives.
- Résumer les tendances observées dans un court rapport.

Travail demandé

1. **Exploration du dataset**
   - Aperçu général : colonnes, types, valeurs manquantes, doublons, distributions.
   - Nettoyage simple si nécessaire (formatage dates, normalisation de chaînes).
2. **Analyse patients & pathologies**
   - Répartition par âge, genre, groupe sanguin.
   - Fréquence des conditions médicales.
   - Analyse des résultats de tests (Normal, Abnormal, Inconclusive).
3. **Analyse hospitalière**
   - Types d'admission (Urgent, Emergency, Elective).
   - Durée moyenne d'hospitalisation (Date d'admission → Date de sortie).
   - Médications les plus courantes.
   - Répartition par hôpital, médecin, assurance.
4. **Analyse financière**
   - Distribution du coût facturé (Billing Amount).
   - Comparaison selon condition médicale, type d'admission ou assurance.
5. **Visualisations obligatoires**
   - Histogrammes, countplots, boxplots (Seaborn/Matplotlib).

    - Visualisations interactives (Plotly).
    - Au moins une visualisation combinant plusieurs dimensions (ex. heatmap croisée, scatter avec couleur par test).
6. **Synthèse**
    - Principales tendances observées : pathologies fréquentes, profils patients, coûts, durée de séjour.
    - Analyse succincte des facteurs liés aux tests anormaux.

Livrables

À déposer sur GitHub :

- Un **notebook Jupyter** complet et propre.
- Un **rapport PDF court** (1–2 pages).
- Une **mini application Streamlit** avec 2–4 graphiques exploratoires.
- Un fichier README.md documentant l'exécution du projet.

Modalités

- Travail individuel ou en équipe de **2 à 4 étudiant·es**.
- Date limite : **10 décembre 2025 – 23h59**.
- Le dépôt GitHub doit être accessible.

Évaluation (100 points)

- Qualité de l'analyse exploratoire. **/25**
- Pertinence et lisibilité des visualisations. **/15**
- Structure et propreté du notebook. **/25**
- Qualité du rapport PDF. **/15**
- Fonctionnalité de l'application Streamlit. **/10**
- Organisation du dépôt GitHub. **/10**

Pénalités

- Mauvais français : −20 %
- Retard : −10 %, puis −5 % par jour (max 7 jours)
- Non-respect des consignes GitHub : −5 à −15 %

Lien du jeu de données : Kaggle