INPT
المعهد الوطني للبريد والمواصلات
Institut National des Postes et Télécommunications

anrt
agence nationale de réglementation des télécommunications
الوكالة الوطنية لتقنين المواصلات

Data Mining et Warehousing

Analyse de l'Évolution des Prix des Cryptomonnaies en Temps Réel

Réalisé par :

AQABLI Souad

EL HANAFI Chaima

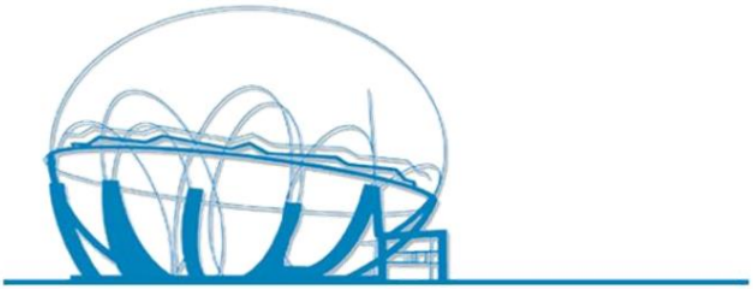
ELKARMI Aya

OUCHEN Imane

SADIKI Chayma

Encadré par :

Mme ELASRI Ikram



INE2 SUD-Cloud et IoT 2024/2025

Table des matières

Introduction

Le marché des cryptomonnaies connaît une croissance rapide et une volatilité importante, attirant autant les investisseurs particuliers que les institutions. Cette dynamique engendre une demande croissante pour des outils d'analyse capables de suivre et d'anticiper les évolutions de prix. Dans ce contexte, les techniques de data mining permettent d'extraire des connaissances utiles à partir de données complexes et massives, tout en fournissant des prédictions sur les comportements futurs des actifs numériques.

Ce projet s'inscrit dans cette perspective, en se concentrant sur l'analyse et la prédiction de l'évolution des prix des cryptomonnaies en temps réel à partir de données extraites directement depuis la plateforme **CoinGecko**. À travers différentes étapes du processus de data mining, nous avons développé un pipeline de traitement allant du scraping à la modélisation prédictive.

Problématique et justification du choix du thème

L'investissement dans les cryptomonnaies repose largement sur l'anticipation des tendances de prix, souvent influencées par des facteurs volatils et peu maîtrisés. Or, les outils classiques d'analyse financière ne permettent pas toujours de capter la complexité des données issues du marché des crypto-actifs.

Dès lors, une problématique centrale se pose : *Comment peut-on, à l'aide des techniques de data mining, analyser et prédire efficacement l'évolution des prix des cryptomonnaies en temps réel, à partir de données extraites du web ?*

Ce thème a été choisi car il représente un cas concret et stimulant d'application du data mining, combinant des défis techniques (scraping, traitement de données temporelles, modélisation prédictive) et des enjeux économiques réels.

Objectifs du projet

L'objectif général de ce projet est de mettre en œuvre une démarche complète de data mining appliquée à l'évolution des cryptomonnaies. Plus précisément, il s'agit de :

- Collecter automatiquement des données actualisées sur les prix des cryptomonnaies via l'API de CoinGecko.
- Comprendre et explorer ces données à travers des analyses statistiques et visuelles.
- Préparer et transformer les données pour une exploitation efficace dans un cadre prédictif.
- Appliquer des techniques de modélisation (régression, séries temporelles, etc.) pour estimer les évolutions futures.
- Évaluer la performance des modèles prédictifs et proposer des pistes d'amélioration.

Méthodologie adoptée

Pour atteindre ces objectifs, le projet s'appuie sur les étapes classiques du processus de data mining :

- **Data selection** : choix de la source (CoinGecko) et des cryptomonnaies à analyser.
- **Data understanding** : compréhension des variables, exploration initiale et visualisation des tendances.
- **Data preparation** : nettoyage, transformation et mise en forme des données temporelles.
- **Modeling** : application de modèles de machine learning pour prédire l'évolution des prix.
- **Evaluation** : comparaison des performances des modèles et discussion des résultats.

Ce rapport présente en détail chaque étape du processus, de la collecte des données à l'interprétation des prédictions, en mettant l'accent sur les techniques utilisées et les enseignements tirés de cette analyse.

1 Sélection des données

1.1 Identification et justification du choix du dataset provenant de CoinGecko

CoinGecko est l'une des principales sources d'informations sur les cryptomonnaies, offrant des données fiables et à jour. Le choix de CoinGecko repose sur plusieurs critères :

- **Accessibilité** : L'API de CoinGecko est gratuite et n'a pas de restrictions strictes, ce qui permet de récupérer une large gamme de données sur les cryptomonnaies.
- **Données complètes** : L'API fournit des informations sur plusieurs centaines de cryptomonnaies, avec des détails comme le prix actuel, la variation sur 24 heures, la capitalisation boursière, etc.
- **Mises à jour fréquentes** : CoinGecko met à jour les données sur les cryptomonnaies à intervalles réguliers, ce qui est essentiel pour les analyses en temps réel ou les prévisions.
- **Simplicité d'utilisation** : L'API est bien documentée et facile à utiliser pour extraire les informations dont nous avons besoin, telles que le prix, le volume des transactions et la variation des prix.

1.2 Code du scraper avec API pour récupérer les données

Afin de collecter des données fiables et à jour sur les cryptomonnaies d'après l'API publique de CoinGecko, nous avons utilisé le script suivant qui permet d'extraire les informations essentielles sur les 500 premières cryptomonnaies classées par capitalisation boursière.

Code du scraper

```python
import requests
import pandas as pd
import json


base_url = "https://api.coingecko.com/api/v3/coins/markets"
vs_currency = 'usd'
coins_per_page = 100
max_pages = 5

cryptos = []

for page in range(1, max_pages + 1):
    params = {
        'vs_currency': vs_currency,
        'order': 'market_cap_desc',
        'per_page': coins_per_page,
        'page': page,
        'sparkline': False
    }

    response = requests.get(base_url, params=params)

    if response.status_code == 200:
        data = response.json()
        for coin in data:
            cryptos.append({
                'name': coin['name'],
                'symbol': coin['symbol'],
                'current_price': coin['current_price'],
                'market_cap': coin['market_cap'],
                'total_volume': coin['total_volume'],
                'price_change_24h_%': coin['price_change_percentage_24h']
```

```
            })
    else:
        print(f"Erreur sur la page {page} : {response.status_code}")
        break

with open('cryptos_500.json', 'w', encoding='utf-8') as f:
    json.dump(cryptos, f, indent=4, ensure_ascii=False)

df = pd.DataFrame(cryptos)
df.to_csv('cryptos_500.csv', index=False, encoding='utf-8')

print(f"{len(cryptos)}cryptos exportées avec succès")
```

1- Importation des bibliothèques nécessaires :

- **requests** : pour envoyer des requêtes HTTP à l'API.
- **pandas** : pour manipuler et exporter les données sous forme de tableau.
- **json** : pour gérer les données au format JSON.

2- Configuration de l'API :

- **base_url** : URL de l'endpoint CoinGecko pour récupérer les données des marchés des cryptomonnaies.
- **vs_currency** = 'usd' : les prix seront exprimés en dollars américains.
- **coins_per_page** = 100 : nombre de cryptos à récupérer par page.
- **max_pages = 5** : nombre total de pages à extraire, soit 500 cryptos au total.

3- Boucle d'extraction paginée : La boucle parcourt les pages de 1 à 5 en configurant à chaque itération les paramètres nécessaires (devise, tri, pagination), puis en envoyant une requête HTTP à l'API via 'requests.get()' pour récupérer les données des cryptomonnaies page par page.

4- Traitement de la réponse et export des données : Pour chaque cryptomonnaie obtenue, les informations clés (nom, symbole, prix, capitalisation, volume, variation) sont extraites et ajoutées à une liste. Cette liste est ensuite sauvegardée en JSON, puis convertie en DataFrame Pandas pour être exportée au format CSV.

2 Compréhension des Données

2.1 Description des Données Collectées

Les données ont été extraites via un scraping réalisé à l'aide de l'API du site **CoinGecko**. Chaque entrée représente les informations clés d'une cryptomonnaie à un instant donné. Voici un exemple d'extrait JSON :

```
[
    {
        "name": "Bitcoin",
        "symbol": "btc",
        "current_price": 82108,
        "market_cap": 1631065775540,
        "total_volume": 45305761744,
        "price_change_24h_%": 1.0877
    },
    {
        "name": "Ethereum",
        "symbol": "eth",
        "current_price": 1556.64,
        "market_cap": 188063081723,
        "total_volume": 19291494644,
        "price_change_24h_%": -0.73754
    },
    ...
```

```
]
```

Listing 2.1 – Extrait de données JSON issues du scraping CoinGecko

2.2 Variables Collectées

- **name** : Nom de la cryptomonnaie (*type : catégorique*).
- **symbol** : Abréviation ou symbole de la cryptomonnaie (*type : catégorique*).
- **current_price** : Prix actuel de la cryptomonnaie (*type : numérique*).
- **market_cap** : Capitalisation boursière (*type : numérique*).
- **total_volume** : Volume total échangé (*type : numérique*).
- **price_change_24h_%** : Variation du prix en pourcentage sur 24 heures (*type : numérique*).

2.3 Défis et Problèmes Potentiels

Durant l'exploration initiale, plusieurs défis ont été identifiés dans les données récupérées :

- **Valeurs manquantes** : Certaines cryptomonnaies peuvent ne pas afficher un prix ou une variation à certains moments.
- **Données dynamiques** : Les prix des cryptomonnaies changent en temps réel, rendant les captures sensibles aux décalages temporels.
- **Volatilité élevée** : Les variations extrêmes des prix peuvent biaiser les analyses ou modèles prédictifs si elles ne sont pas traitées correctement.
- **Valeurs indéterminées** : la valeur infinie ou une division par 0.
- **Valeurs aberrantes** : Des valeurs extrêmes peuvent exister dans les champs comme `current_price` ou `price_change_24h_%`.

La compréhension des données permet de préparer les prochaines étapes du pipeline de Data Mining, notamment la phase de nettoyage et de préparation. Ces premières observations seront essentielles pour définir des stratégies de gestion des données problématiques et concevoir des modèles robustes.

3 Préparation des données

3.1 Code de nettoyage et préparation des données

Afin d'utiliser les données extraites, il est nécessaire de les traiter pour assurer leur qualité et leur cohérence. Cela inclut la gestion des doublons, des valeurs manquantes, ainsi que des valeurs aberrantes. Voici le code de prétraitement appliqué aux données extraites des cryptomonnaies, qui inclut ces étapes.

```
import pandas as pd
import numpy as np

with open('cryptos_with_problems.json', 'r', encoding='utf-8') as f:
    cryptos = json.load(f)

# Convertir en DataFrame
df = pd.DataFrame(cryptos)
# 1. Vérification des doublons
print("Nombre de doublons avant suppression:", df.duplicated().sum())
df = df.drop_duplicates()  # Supprimer les doublons
print("Nombre de doublons après suppression:", df.duplicated().sum())
# 2. Vérification des valeurs manquantes (NaN)
print("Valeurs manquantes avant traitement:")
print(df.isnull().sum())
# Imputer les valeurs manquantes (par la moyenne )
df['current_price'] = df['current_price'].fillna(df['current_price'].mean())
df['market_cap'] = df['market_cap'].fillna(df['market_cap'].mean())
df['total_volume'] = df['total_volume'].fillna(df['total_volume'].mean())
df['price_change_24h_%'] = df['price_change_24h_%'].fillna(df['
```

```
    price_change_24h_%'].mean())
print("Valeurs manquantes après traitement:")
print(df.isnull().sum())
# 3. Vérification des valeurs aberrantes : ici, on peut supposer qu'un prix
    ou une capitalisation négatifs sont des anomalies
print("Vérification des valeurs négatives dans 'current_price' :")
print(df[df['current_price'] < 0])
# Remplacer les prix négatifs par la moyenne des prix
df['current_price'] = df['current_price'].apply(lambda x: x if x >= 0 else
    df['current_price'].mean())
# 4. Vérification des valeurs aberrantes pour la capitalisation boursière
print("Vérification des valeurs aberrantes dans 'market_cap' :")
print(df[df['market_cap'] < 0])
# Remplacer les capitalisations boursières négatives par la moyenne des
    capitalisations
df['market_cap'] = df['market_cap'].apply(lambda x: x if x >= 0 else df['
    market_cap'].mean())
# 5. Vérification des outliers : Utilisation de l'écart interquartile (IQR)
Q1 = df['price_change_24h_%'].quantile(0.25)
Q3 = df['price_change_24h_%'].quantile(0.75)
IQR = Q3 - Q1
# Détection des outliers
outliers = df[(df['price_change_24h_%'] < (Q1 - 1.5 * IQR)) | (df['
    price_change_24h_%'] > (Q3 + 1.5 * IQR))]
print(f"Nombre d'outliers détectés dans 'price_change_24h_%': {len(outliers)
    }")
# Supprimer les outliers
df = df[~df['price_change_24h_%'].isin(outliers['price_change_24h_%'])]
with open('cryptos_preprocessed.json', 'w', encoding='utf-8') as f:
    json.dump(df.to_dict(orient='records'), f, indent=4, ensure_ascii=False)
df.to_csv('cryptos_preprocessed.csv', index=False, encoding='utf-8')
print("Données après prétraitement :")
print(df.head())
```

3.2 Explication des étapes suivies

a. Chargement des Données

Le jeu de données utilisé provient de l'API CoinGecko, qui fournit des informations détaillées sur les cryptomonnaies, notamment leur nom, symbole, prix actuel, capitalisation boursière, volume total des transactions et variation des prix sur 24 heures. Les données ont été récupérées et sauvegardées au format JSON pour permettre une analyse ultérieure.

b.Vérification des Doublons

Avant de traiter les données, nous avons vérifié la présence de doublons, car ceux-ci peuvent perturber l'analyse. Grâce à la fonction duplicated() de pandas, nous avons identifié et supprimé les lignes dupliquées pour garantir que chaque cryptomonnaie soit unique dans le jeu de données.

c. Gestion des Valeurs Manquantes

Certaines colonnes de notre dataset contenaient des valeurs manquantes (NaN). Pour les traiter, nous avons imputé ces valeurs par la moyenne des colonnes correspondantes, une méthode courante qui permet de préserver la distribution des données sans introduire de biais significatif.

Détection et Traitement des Outliers

Les outliers, ou valeurs aberrantes, peuvent fausser les analyses et les prédictions. Nous avons utilisé l'écart interquartile (IQR) pour identifier les outliers dans la colonne price_change_24h. Les valeurs en dehors de 1,5 fois l'IQR ont été considérées comme aberrantes et supprimées.

Exportation des Données Prétraitées

Après prétraitement, les données ont été exportées au format JSON et CSV. Le format JSON est pratique pour la manipulation en Python, tandis que le CSV permet une utilisation facile dans des outils comme Excel.

Le prétraitement a permis d'assurer la qualité des données en éliminant les doublons, en gérant les valeurs manquantes et négatives, et en traitant les outliers. Ces données sont désormais prêtes pour l'analyse ou l'entraînement de modèles de machine learning, assurant ainsi leur fiabilité et représentativité.

4 L'analyse exploratoire des données (AED)

Introduction

Avant de construire un modèle prédictif ou de prendre des décisions basées sur des données, il est essentiel de bien comprendre la structure, les caractéristiques et les relations internes des données. C'est précisément l'objectif de l'analyse exploratoire des données, souvent appelée AED ou EDA (Exploratory Data Analysis) en anglais.

Dans le contexte de notre étude des cryptomonnaies, l'AED nous permet de :

- détecter les tendances générales du marché,
- comprendre la distribution de variables importantes comme le prix, le volume ou la capitalisation boursière,
- évaluer les relations entre les variables numériques via des corrélations,
- observer les variations temporelles pour certaines cryptomonnaies clés,
- et repérer les actifs les plus volatils, ce qui peut avoir un impact direct sur les stratégies d'investissement.

À travers cette analyse, nous visons à obtenir une vision claire et intuitive des données collectées, à l'aide de représentations graphiques et de mesures statistiques. Ces observations formeront une base solide pour les étapes suivantes de notre projet, notamment la modélisation et la prédiction.

4.1 Analyse de la distribution des prix

À l'aide du code Python suivant, nous avons pu tracer un graphique illustrant la répartition des cryptomonnaies en fonction de leur prix approximatif. Ce graphique permet de visualiser la concentration des

cryptomonnaies selon leur valeur en dollars américains.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import json

# Load JSON data
with open('cryptos_500.json', 'r') as file:
    data = json.load(file)

# Convert to DataFrame
df = pd.DataFrame(data)
plt.figure(figsize=(8,5))
sns.histplot(df['current_price'], bins=50)
plt.title("Distribution of Current Prices")
plt.xlabel("Price (USD)")
plt.ylabel("Count")
plt.show()
```

Voici le graphique obtenu :

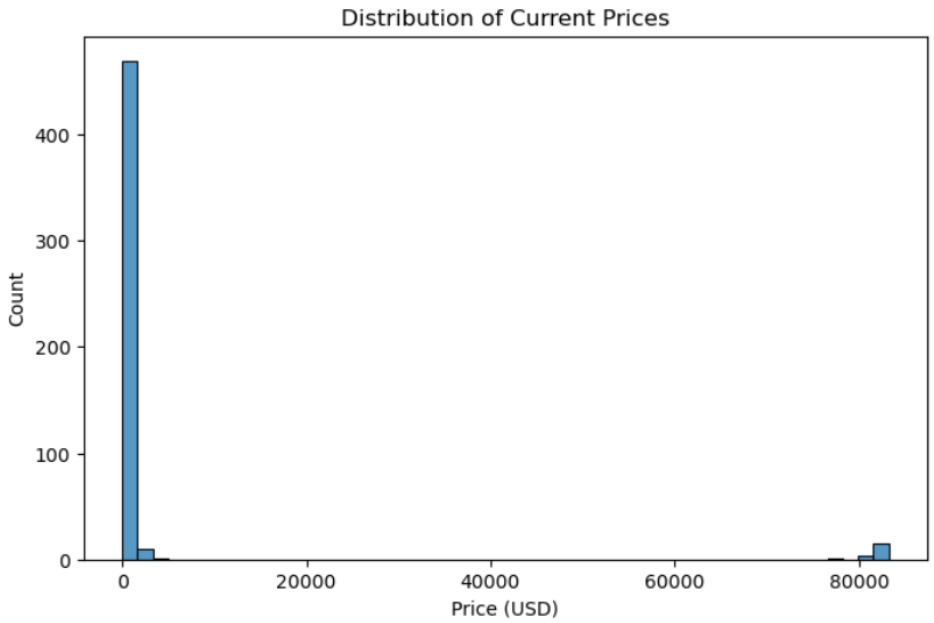


FIGURE 4.1 – Répartition des prix actuels des cryptomonnaies

Le graphique ci-dessus illustre les prix actuels des principales cryptomonnaies en dollars américains. On observe que seules quelques cryptomonnaies (moins de 20) présentent un prix élevé, tandis que la majorité des cryptomonnaies se situent dans une fourchette de prix relativement basse. En effet, on remarque une concentration importante de cryptomonnaies dont le prix est inférieur à quelques dollars, avec une fréquence dépassant 400 occurrences. Cette distribution montre une asymétrie marquée, suggérant que le marché des cryptomonnaies est dominé en nombre par des actifs à faible valeur unitaire.

4.2 Analyse Exploratoire des Données (EDA)

4.2.1 Objectif de l'Analyse

Dans cette section, nous effectuons une analyse exploratoire des données (EDA) pour mieux comprendre les relations et les caractéristiques des cryptomonnaies contenues dans notre dataset. Cette analyse inclut des visualisations ainsi que des calculs de corrélations entre différentes variables. Cela nous permettra de tirer des insights utiles pour affiner nos futurs modèles de machine learning.

4.2.2 Corrélations entre les Variables

À l'aide du code Python suivant, Une heatmap des corrélations entre les différentes variables a été générée pour mieux comprendre les relations entre celles-ci.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Extraire uniquement les colonnes numériques
numeric_df = df[['current_price', 'market_cap', 'total_volume', '
   price_change_24h_%']]

# Calcul des corrélations
correlation_matrix = numeric_df.corr()

# Affichage de la heatmap
plt.figure(figsize=(8, 6))
```

```
sns.heatmap(correlation_matrix, annot=True, cmap='YlGnBu', fmt=".2f")

plt.title("Corrélations entre les variables numériques")
plt.tight_layout()
plt.show()
```

Voici le Heatmap obtenu :

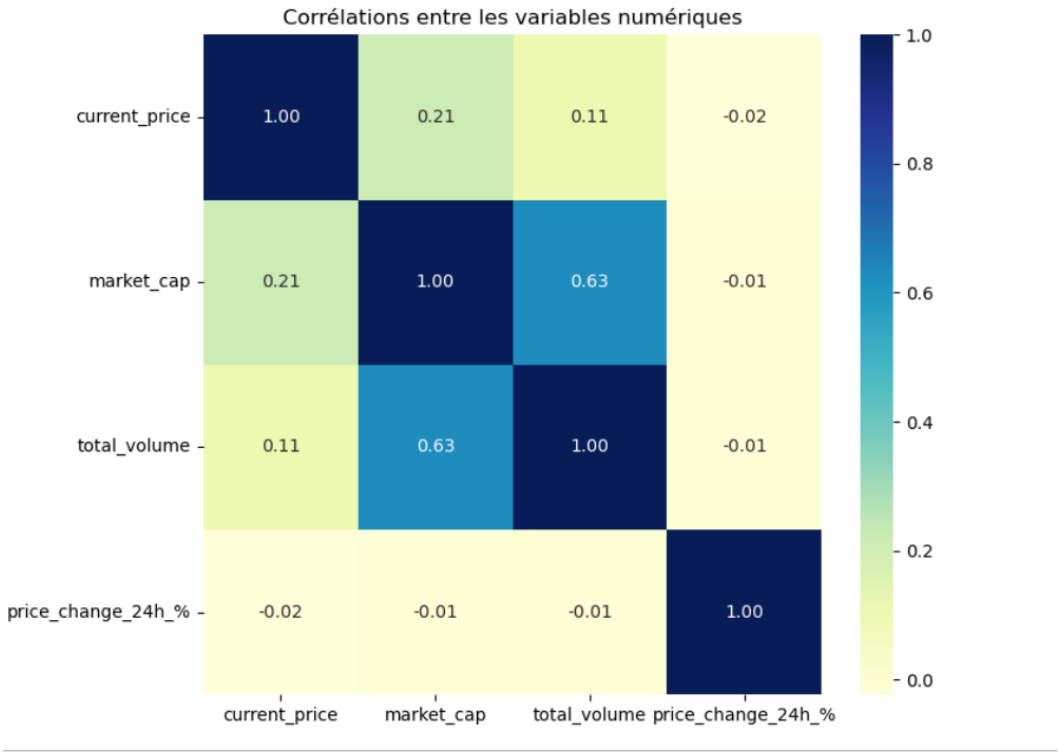


FIGURE 4.2 – Heatmap des Corrélations

4.2.3 interprétation du Heatmap

— La corrélation entre `current_price` et `market_cap` est de **0.21**. Cela indique une faible relation positive, suggérant que le prix actuel d'une cryptomonnaie n'est que modérément lié à sa capitalisation boursière. Les cryptomonnaies ayant un prix élevé n'ont pas nécessairement une capitalisation boursière élevée.

— La corrélation entre `current_price` et `total_volume` est de **0.11**, ce qui indique une très faible relation positive. Le prix actuel d'une cryptomonnaie n'est donc que très faiblement lié à son volume total échangé.

— La corrélation entre `market_cap` et `total_volume` est de **0.63**. Cela montre une relation modérée,

suggérant qu'une capitalisation boursière élevée est souvent associée à un volume d'échanges élevé, bien qu'il puisse y avoir des exceptions.

— Les corrélations entre `price_change_24h_%` et les autres variables sont très faibles, avec des valeurs de **-0.02** ou **-0.01**, indiquant que les variations de prix à court terme sont peu influencées par les caractéristiques fondamentales des cryptomonnaies.

Ces corrélations nous permettent de mieux comprendre les relations sous-jacentes entre les différentes variables du dataset. Elles nous aident à déterminer quels facteurs influencent le plus les fluctuations des cryptomonnaies et à identifier les variables les plus pertinentes pour nos futurs modèles de machine learning.

4.2.4 Analyse des variations temporelles des cryptomonnaies

Afin d'observer le comportement des cryptomonnaies sur une période donnée, nous avons récupéré les données de prix à intervalles réguliers depuis la plateforme CoinGecko. Les graphiques suivants représentent les variations temporelles des cryptomonnaies les plus importantes de notre dataset.

Voici les courbes de variation :

FIGURE 4.3 – Variation temporelle de Bitcoin

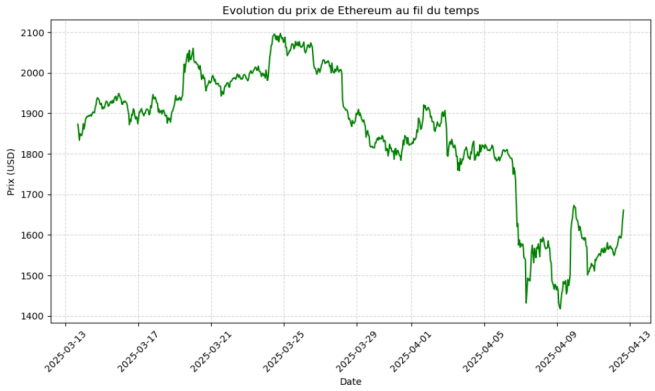


FIGURE 4.4 – Variation temporelle d'Ethereum

FIGURE 4.5 – Variation temporelle de Wrapped Bitcoin

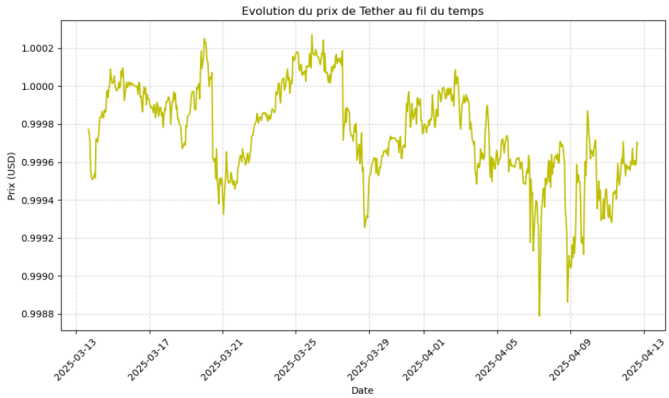


FIGURE 4.6 – Variation temporelle de Tether (USDT)

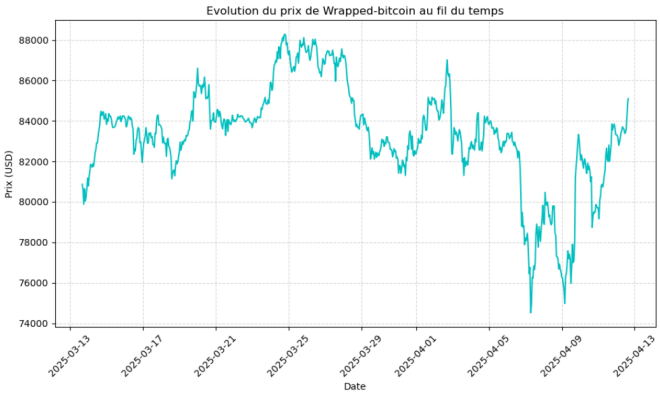


FIGURE 4.7 – Variation temporelle de Wrapped Bitcoin

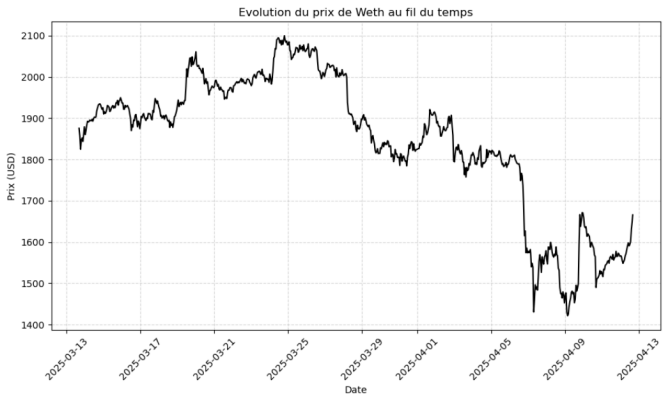


FIGURE 4.8 – Variation temporelle de WETH

Analyse et interprétation

L'analyse de ces courbes met en évidence une **corrélation notable** entre les cryptomonnaies suivantes : **Bitcoin**, **Ethereum**, **Wrapped Bitcoin**, **Tether (USDT)** et **WETH**. En effet, ces cryptomonnaies présentent des tendances similaires dans leurs fluctuations de prix, indiquant qu'elles réagissent souvent de manière cohérente face aux événements du marché.

Cette synchronisation suggère une interdépendance ou un comportement similaire des investisseurs vis-à-vis de ces actifs.

Cependant, certaines cryptomonnaies comme **Solv Protocol SolvBTC** ne suivent pas ce même schéma. Leurs courbes ne présentent pas de corrélation claire avec les autres, ce qui indique un comportement de marché plus isolé.

Cette observation est utile dans le cadre d'une stratégie de diversification de portefeuille ou d'analyse prédictive des mouvements de marché.

4.2.5 Analyse de la volatilité des cryptomonnaies

À l'aide du code Python suivant, nous avons analysé les variations de prix sur 24 heures pour identifier les cryptomonnaies les plus volatiles.

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Charger les données
df = pd.read_json("cryptos_500.json")


# Trier par variation de prix en 24h (en pourcentage absolu)
df['abs_price_change_24h'] = df['price_change_24h_%'].abs()
top_volatile = df.sort_values(by='abs_price_change_24h', ascending=False).
    head(10)

# Affichage
print("Cryptos les plus volatiles en 24h :")
print(top_volatile[['name', 'price_change_24h_%']])
```

Le tableau suivant présente les 10 cryptomonnaies ayant enregistré les plus grandes variations en pourcentage :

Nom de la cryptomonnaie	**Variation en 24h (%)**
Onyxcoin	+55.68
Orca	+53.00
SWFTCOIN	+46.45
XYO Network	+41.28
Zircuit	+38.28
Popcat	+29.99
Aergo	+28.84
Fartcoin	+25.21
Alchemist AI	+22.21
Tribe	**-19.57**

TABLE 4.1 – Cryptomonnaies les plus volatiles sur 24h

— Ces cryptos sont particulièrement sensibles aux mouvements du marché.

— Une forte volatilité peut signifier des opportunités de profit à court terme, mais aussi un risque accru.

5 Modélisation et Évaluation des Modèles

Les principales responsabilités de cette section incluent :

- La construction et l'entraînement de modèles de **régression** (prédiction de prix) et de **classification** (prédiction de la tendance du prix).
- L'évaluation des performances des modèles à l'aide de métriques appropriées (*Accuracy*, *Precision*, *Recall*, *RMSE*, R^2, etc.).
- Le réglage des hyperparamètres et la sélection de caractéristiques pertinentes.
- La comparaison des performances des différents algorithmes pour retenir les plus efficaces.

Il est important de souligner qu'un même jeu de données peut être exploité de plusieurs manières selon les besoins métiers spécifiques d'une entreprise ou d'une organisation. Dans notre cas, nous avons choisi d'explorer deux axes d'exploitation :

- **Classification binaire : Tendance du prix**

  L'objectif ici est de prédire si le prix d'une cryptomonnaie va augmenter ou diminuer dans les prochaines 24 heures. Ce type de modèle peut être utile pour des plateformes de trading souhaitant alerter les utilisateurs sur la tendance à venir.

- **Régression : Prédiction du prix**

  Ce modèle vise à estimer la valeur exacte du prix d'une cryptomonnaie à partir de variables comme la capitalisation boursière, le volume échangé, ou la variation de prix. Une telle approche peut être exploitée pour la gestion de portefeuilles d'actifs ou la valorisation dynamique de cryptos.

5.1 Classification binaire : Tendance du prix (hausse ou baisse)

5.1.1 Préparation des données

Nous avons dérivé une variable cible `price_up`, valant 1 si la variation du prix sur 24h est positive, et 0 sinon. Trois variables ont été retenues comme prédicteurs : `market_cap`, `total_volume` et `current_price`. Les données ont été standardisées avant l'entraînement.

5.1.2 Modèles utilisés

Trois algorithmes de classification ont été comparés :

— Régression logistique
— Forêt aléatoire (Random Forest)
— Gradient Boosting

5.1.3 Évaluation des performances

Les performances ont été comparées à l'aide des métriques suivantes : **Accuracy**, **Precision**, et **Recall**.

	Model	Accuracy	Precision	Recall
0	Logistic Regression	0.660000	0.668919	0.980198
1	Random Forest	0.686667	0.775510	0.752475
2	Gradient Boosting	0.666667	0.733945	0.792079

FIGURE 5.1 – Résumé des performances des modèles de classification

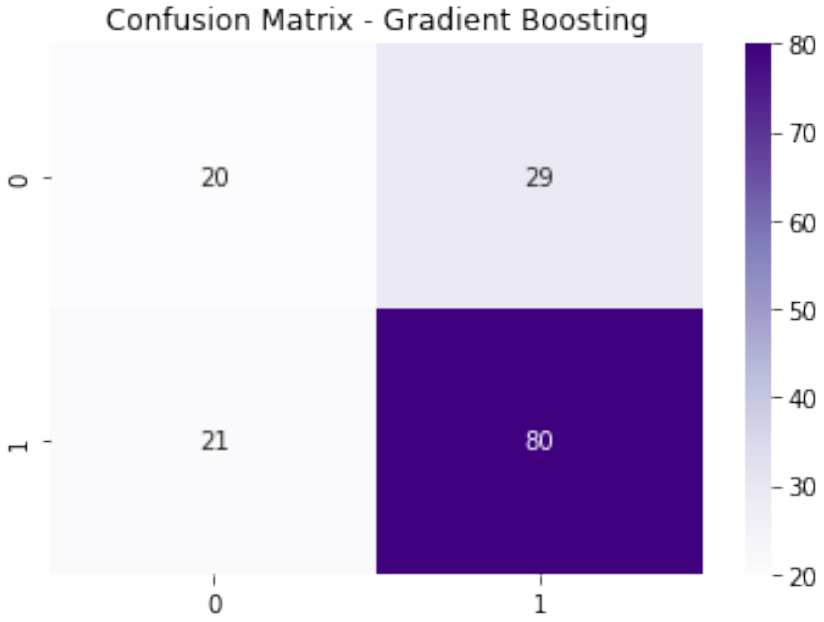


FIGURE 5.2 – Matrice de confusion – Modèle Gradient Boosting

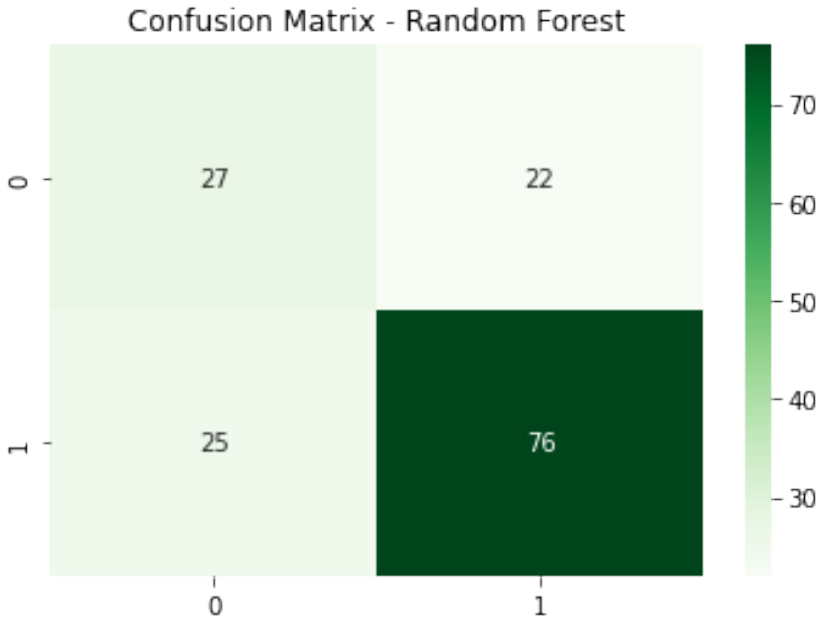


FIGURE 5.3 – Matrice de confusion – Modèle Random Forest

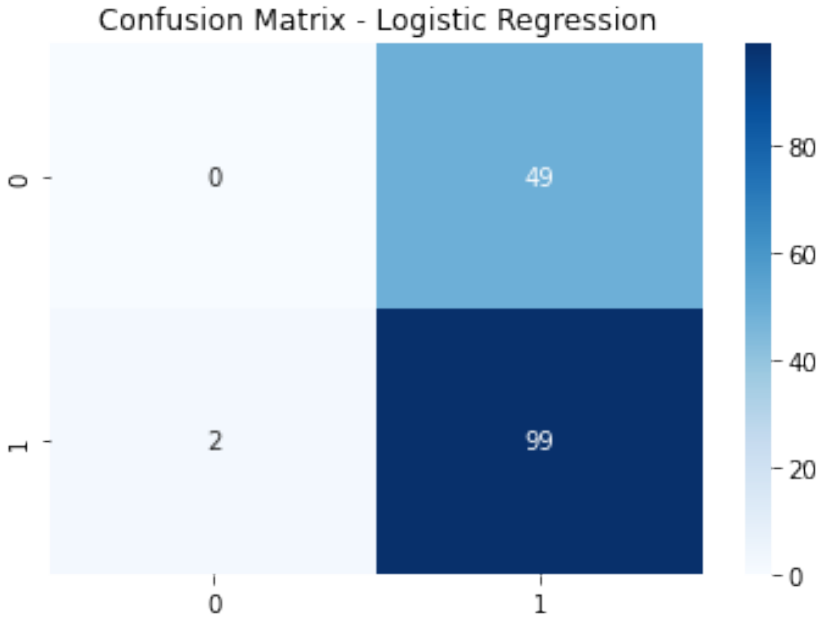


FIGURE 5.4 – Matrice de confusion – Modèle Regression Logiqtique

5.1.4 Exploration visuelle des données

Le graphe généré est un *pairplot* réalisé à l'aide de la bibliothèque Seaborn, qui permet de visualiser la relation entre plusieurs variables simultanément. Dans ce cas, le *pairplot* inclut quatre caractéristiques : la capitalisation boursière (`market_cap`), le volume total échangé (`total_volume`), le prix actuel (`current_price`) et la variable cible binaire (`price_up`), qui indique si le prix d'une cryptomonnaie a augmenté ou diminué sur les dernières 24 heures.

Le *pairplot* permet d'examiner les corrélations et les distributions des différentes caractéristiques du dataset. Chaque graphique de cette matrice montre la relation entre deux variables, tandis que les graphiques diagonaux affichent les distributions individuelles des variables. L'utilisation du paramètre `hue='price_up'` permet de colorier les points en fonction de la cible, facilitant ainsi l'analyse visuelle des tendances.

Les couleurs assignées *(rouge pour les valeurs de* `price_up` *égales à 0 et vert pour celles égales à 1)* permettent de distinguer clairement les cryptomonnaies dont le prix a baissé de celles dont le prix a augmenté. Ce type de graphique est particulièrement utile pour identifier des patterns ou des relations entre les variables qui pourraient influencer la tendance des prix.

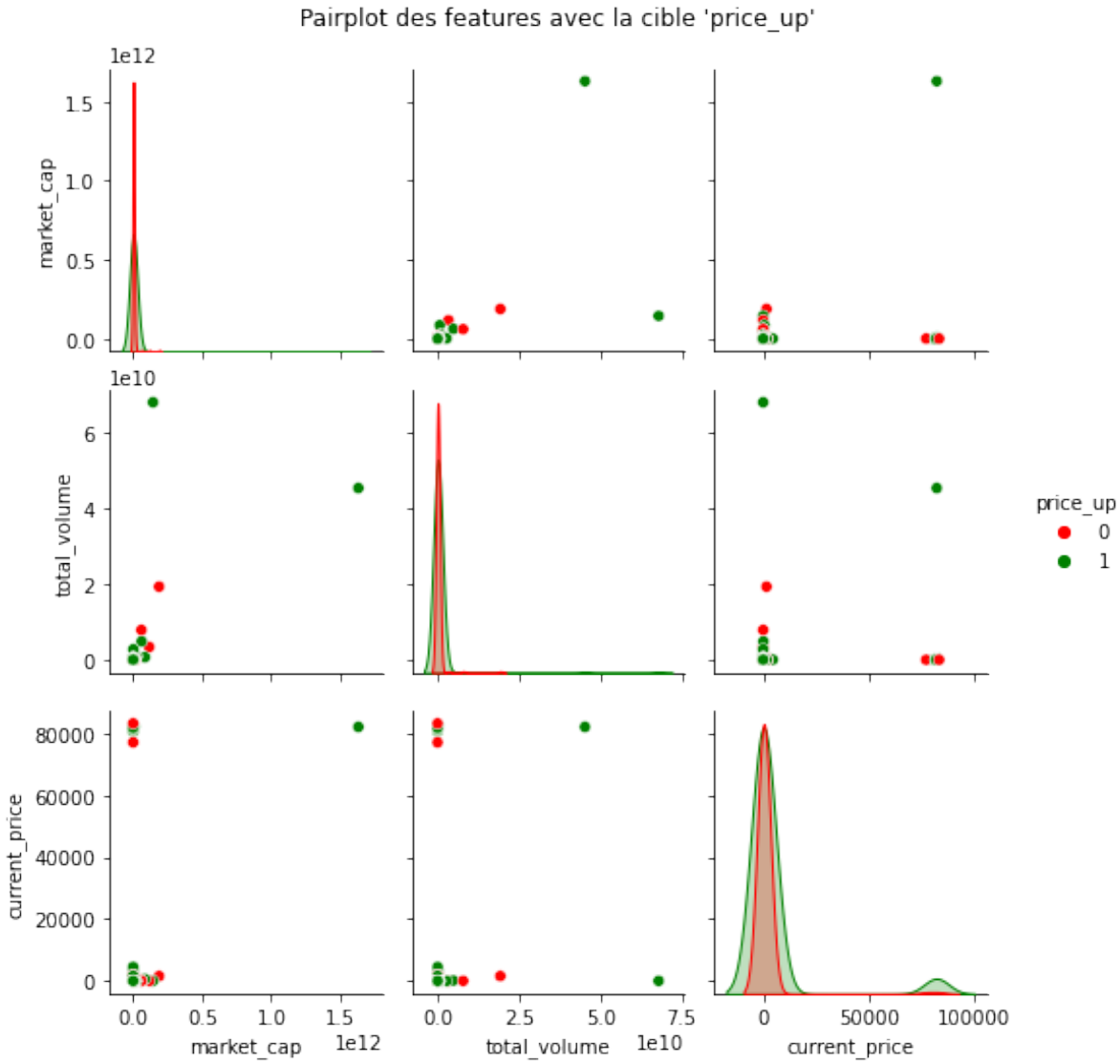


FIGURE 5.5 – Pairplot des features en fonction de la variable cible

5.2 Régression : Prédiction du prix actuel

5.2.1 Préparation des données

Pour cette tâche de régression, l'objectif est de prédire le `current_price` d'une cryptomonnaie à partir de trois variables explicatives : `market_cap`, `total_volume` et `price_change_24h_%`. Après avoir chargé les données, les valeurs manquantes, infinies ou non définies ont été supprimées. Les données ont ensuite été divisées en ensembles d'entraînement (70%) et de test (30%).

5.2.2 Modèles utilisés

Trois modèles de régression ont été implémentés pour cette tâche :

— Régression linéaire

— Forêt aléatoire (Random Forest Regressor)

— Gradient Boosting Regressor

5.2.3 Évaluation des performances

Les modèles ont été évalués à l'aide des métriques suivantes :

— **RMSE** (Root Mean Squared Error)

— **MAE** (Mean Absolute Error)

— **R²** (coefficient de détermination)

Un résumé des résultats obtenus est présenté dans le tableau 5.6. Ces métriques permettent de comparer la précision et la robustesse des différents modèles pour la prédiction du prix.

	Model	RMSE	MAE	R²
0	Linear Regression	17781.847730	6457.098244	-0.222272
1	Random Forest	15685.113914	6168.962120	0.048980
2	Gradient Boosting	19333.248622	7757.864244	-0.444854

FIGURE 5.6 – Résumé des performances des modèles de régression

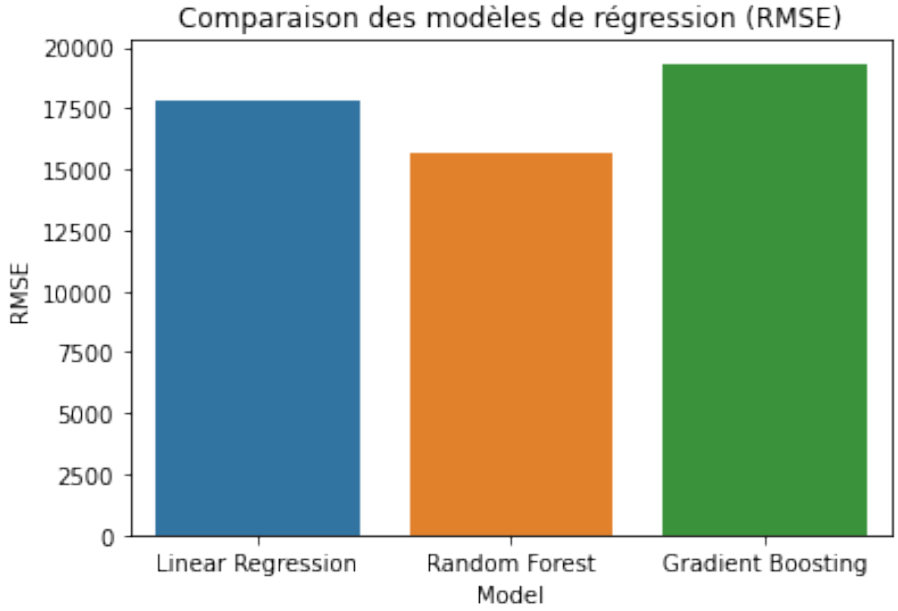


FIGURE 5.7 – Comparaison des modèles de régression selon la métrique RMSE

5.2.4 Exploration visuelle des données

Avant la modélisation, nous avons examiné la distribution de la variable `price_change_24h_%`, qui représente la variation relative du prix sur les dernières 24 heures. Cette analyse permet de mieux comprendre la dynamique du marché et d'évaluer la présence de valeurs extrêmes ou de biais.

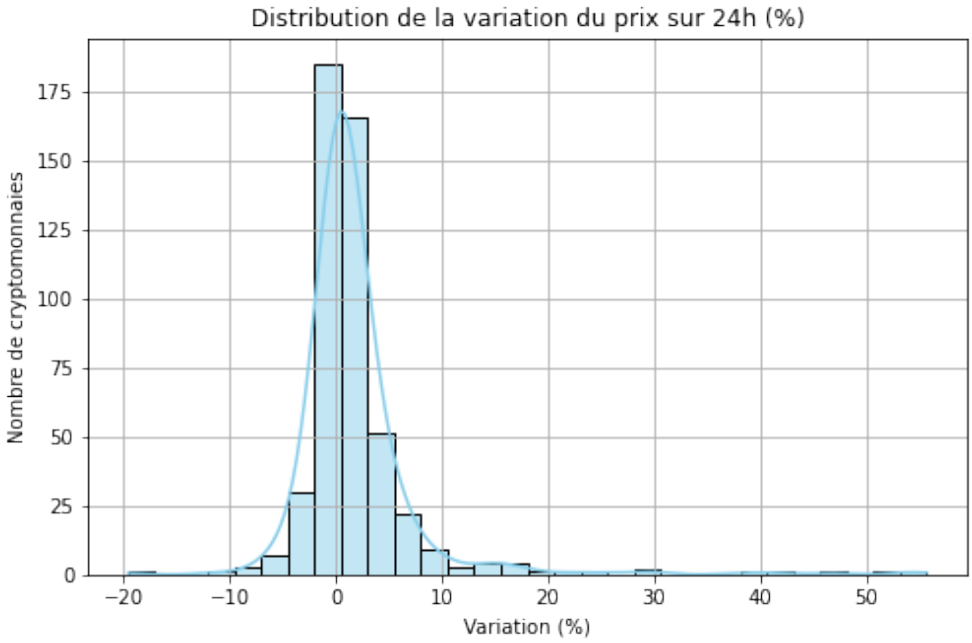


FIGURE 5.8 – Distribution de la variation du prix sur 24h (%)

Conclusion

Ce projet a permis de concevoir et de mettre en œuvre une solution complète de suivi et de prédiction des prix des cryptomonnaies en temps réel, en mobilisant à la fois des techniques de *web scraping*, de préparation et d'analyse de données, ainsi que des modèles de *machine learning* adaptés aux problématiques de séries temporelles.

À travers l'exploitation des données extraites de *CoinGecko* via son API, nous avons pu observer la dynamique du marché des cryptomonnaies, caractérisé par une volatilité importante et des fluctuations continues. Cette nature instable a représenté un défi, mais aussi une opportunité pour appliquer des méthodes avancées de prévision (régression) et de classification des tendances.

L'ensemble du projet a suivi une démarche rigoureuse : sélection et compréhension des données, nettoyage, création de nouvelles variables pertinentes, visualisation exploratoire, modélisation et évaluation. Les modèles de régression ont permis d'estimer avec une certaine précision l'évolution future des prix, tandis que les modèles de classification ont fourni une vision synthétique de la tendance (hausse, baisse, stabilité) à court terme.

Ce travail a mis en lumière plusieurs enseignements :

- L'importance cruciale de la qualité et de la fraîcheur des données dans les systèmes temps réel.
- La nécessité d'un traitement adapté des *outliers* et des valeurs manquantes dans des données aussi volatiles.
- La complémentarité des approches exploratoires et prédictives pour extraire de la valeur des données.

Malgré les limites inhérentes aux données disponibles (notamment la fréquence d'échantillonnage ou la couverture historique limitée), les résultats obtenus sont encourageants et ouvrent des perspectives d'amélioration. Des extensions possibles incluent l'ajout de sources de données externes (actualités, volumes de transaction, réseaux sociaux), l'utilisation de modèles séquentiels plus avancés (comme les

RNN ou *LSTM*), ou encore la mise en place d'un système d'alerte automatique basé sur les prédictions.

En somme, ce projet constitue une expérience riche et complète qui illustre les apports concrets du *Data Mining* et du *Machine Learning* dans un domaine aussi complexe et dynamique que celui des cryptomonnaies. Il a également renforcé notre capacité à collaborer efficacement en équipe, à intégrer plusieurs compétences techniques, et à produire une solution cohérente, automatisée et orientée résultats.