



# Projet OPA

## CryptoBot avec Binance

Oualid Lemtine 

Samuel Vicat 

Tuteur : Simon Cariou 



DataScientest



Nov23\_Continu\_de - 2024

# SOMMAIR

E

1

Présentation du projet

2

Étapes importantes du projet

3

Focus Bdd

4

Machine Learning

5

Conclusion



# Présentation du projet

Le projet OPA a pour objectif de créer un **bot de trading** pour investir sur le marché des crypto monnaies via la plateforme Binance.

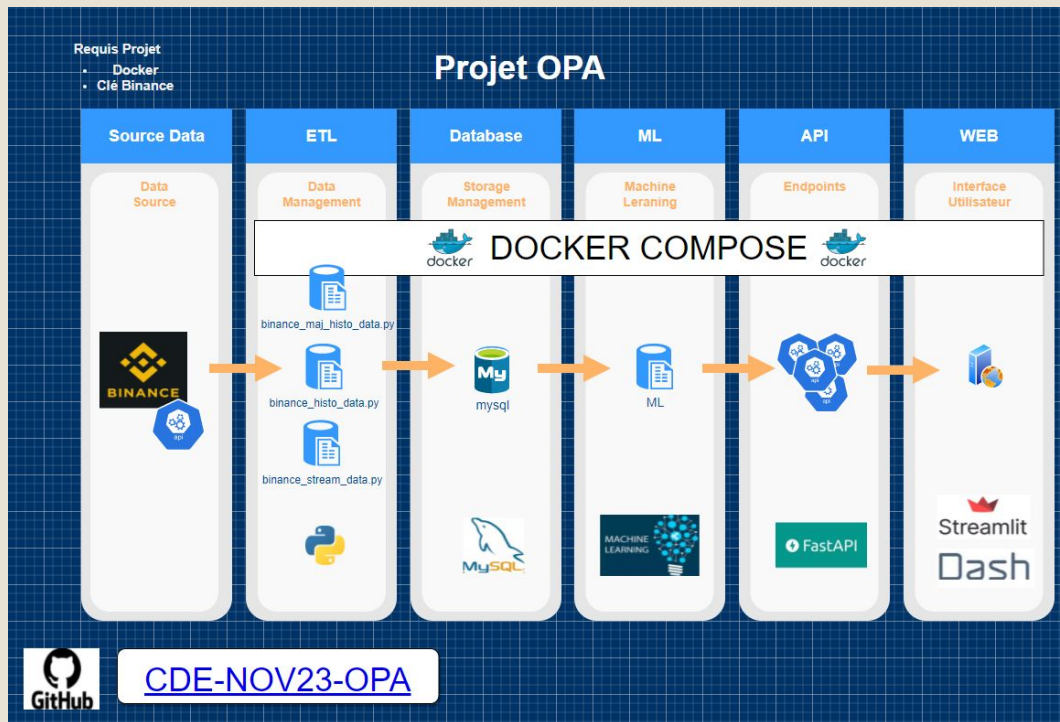
A l'aide des méthodes de machine learning, nous allons développer des stratégies d'achat/vente et industrialiser leur déploiement.



<https://github.com/DataScientest-Studio/CDE-NOV23-OPA>



# Architecture



# Étape : Récolte et Architecture de la donnée

## Via la clé Binance

récupération des données Crypto sur l'API  
Binance

Aide: [Comment créer des clés API sur Binance ?](#)

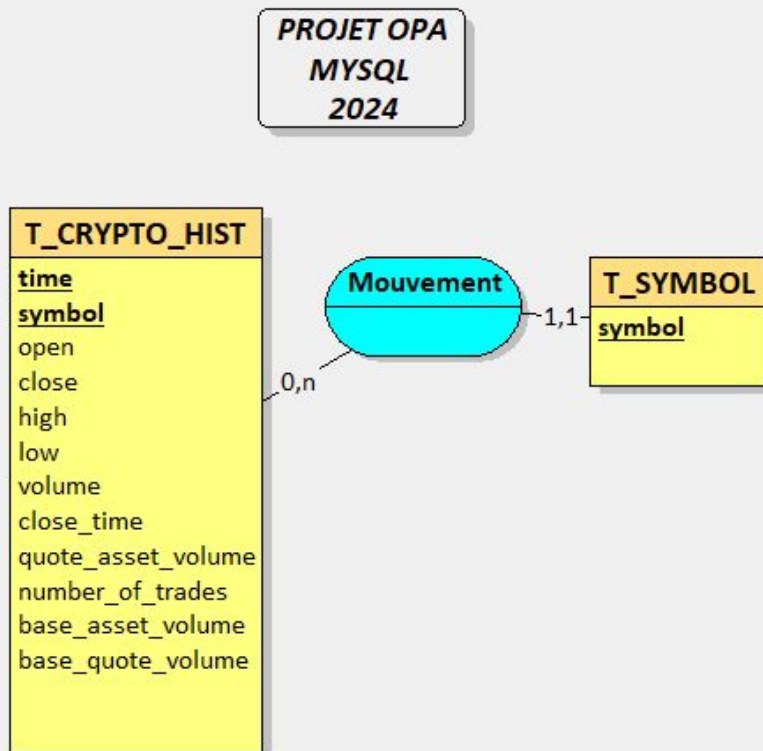
## Base de donnée Mysql

### Pourquoi Mysql

- Base de donnée reconnu et populaire
- Les données sont structurés, avec un format stable
- Facilite l'entraînement des données pour le ML



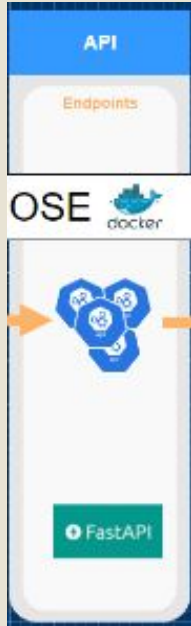
# Schema Table



A noter:  
Les données de streaming sert directement dans le Dash



# Étapes : Déploiement API



Pourquoi Fast API :

- Pour sécuriser la base de donnée
- Permettre d'afficher les données et recommandations sur différents point d'accès
- Gérer les mises à jours de la base



# Étapes : Déploiement ML



Régression : Prédiction de prix

Classification:





# Conclusion

L'application Projet OPA rend le service demandé mais à besoin d'être optimisé pour une bonne utilisation plus complète

## Axes d'amélioration :

### - MySQL :

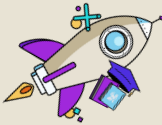
- Optimisation de la base : Ajout d'un index
- Gestion de la mise à jour des données
  - Définir une plage de mises à jour
  - En cas de crash

### - FastAPI

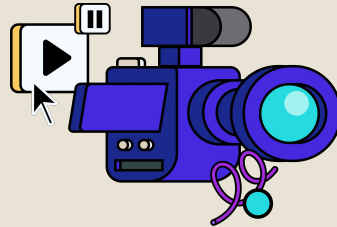
- Amélioration de la sécurisation
- Gestion des droits administrateurs et utilisateurs suivants les tâches

### - Dash

- Mise en place d'indicateur KPI (ex: Sharpe\*, marge)
- Interface avec login utilisateur pour avoir des vues et API personnalisées



# Démonstration



# DES QUESTIONS ?

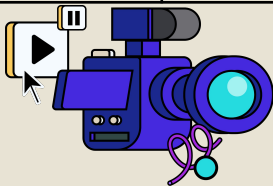


# **Annexe**



# Étapes 1 : Récolte des données

Etape	Description	Objectif	Modules / Masterclass / Templates	Conditions de validation du projet
1	Récolte des données	<p>Récolter deux types de données en passant par l'API Binance en passant par une architecture de streaming.</p> <ol style="list-style-type: none"><li>Grâce à cette API, on peut aller récupérer des informations sur les cours des différents marchés (BTC-USDT, BTC-ETH, ...). Le but sera de créer une fonction de récupération de données générique afin de pouvoir avoir les données de n'importe quel marché. Il faudra aussi créer un script de pré-processing pour réorganiser les données sortant du streaming afin qu'elles soient propres.</li><li>Récupérer les données historiques, pré-processées pour pouvoir entraîner notre futur modèle</li></ol>		<p>Fichier explicatif du traitement (doc / pdf)</p> <p>Un fichier json d'exemple de récupération.</p>

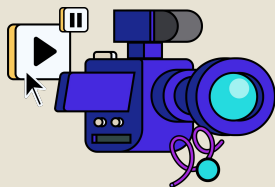


## Présentation



## Étapes 2 : Architecture de la donnée

Etape	Description	Objectif	Modules / Masterclass / Templates	Conditions de validation du projet
2	Architecture de la donnée	<p>Il s'agit de choisir la solution de stockage la plus adaptée:</p> <ul style="list-style-type: none"><li>- 2 tables SQL, une pour les données historiques et une autre pour les données streaming.</li><li>- Une DB Mongo/Elastic comportant 2 collections: une pour les données stream et une autre pour les données historiques.</li></ul>	<p>142 - SQL (Architecture des données)</p> <p>Elasticsearch</p> <p>143 - MongoDB</p>	<p>Une base de données relationnelle</p> <p>Un fichier de requête SQL pour montrer que c'est bien fonctionnel</p> <p>Même rendu mais exemples de requêtes Elastic/Mongo</p>

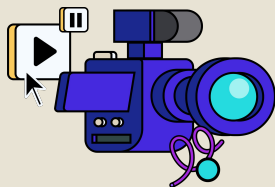


Présentation



## Étapes 3 : Consommation de la donnée

Etape	Description	Objectif	Modules / Masterclass / Templates	Conditions de validation du projet
3	Consommation de la donnée	Implémenter un Dashboard qui permettrait de requêter sur la bdd streaming pour suivre une évolution en direct du cours <b>OU</b> utiliser un algo de Machine Learning appliqué à la finance qui permettra de retourner une décision d'achat ou non. (Quelle stratégie pour la vente ?) Aller plus loin: prédiction de gains.	ElasticSearch, Kibana, Dash (module complémentaire) DE120 - ML pour les DE DE121	Notebook ou tableau de bord Dash

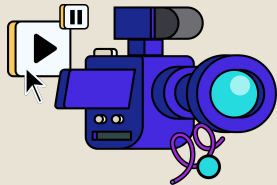


## Présentation



## Étapes 4 : Déploiement

Etape	Description	Objectif	Modules / Masterclass / Templates	Conditions de validation du projet
4	Déploiement	<p>Faire une API pour:</p> <ul style="list-style-type: none"><li>- <u>Si choix du dashboard</u>: Être capable de requêter des données dans les BDDs créées. Ex: Le cours actuel de la paire de trading, les prix moyens au cours des dernières années, l'évolution du cours de la monnaie par rapport à une moyenne mobile... aucune limite sur les données que vous souhaiteriez mettre en avant</li><li>- <u>Si choix de ML</u>: Mettre en avant les prédictions du modèle de ML</li></ul> <p>Implémenter des tests unitaires pour tester votre API.</p>	FastAPI, Flask	Fichiers de l'API
		<p>Dockeriser tout le projet pour qu'il soit reproductible sur n'importe quel machine</p> <p>Docker-Compose des différents conteneurs + réseau + ports pour API.</p>	Docker	<p>Fichier Yaml du docker-compose</p> <p>Possible de faire un setup.sh pour la création des images</p>



## Présentation





## Étapes 5 : Automatisation des flux (facultatif)

Etape	Description	Objectif	Modules / Masterclass / Templates	Conditions de validation du projet
5	Automatisation des flux (facultatif)	Il faudra requêter l'API quotidiennement via Airflow. Nous ne pouvons pas utiliser Airflow pour les données en streaming, mais vous pouvez tester Nifi.	Airflow, (Nifi)	Fichier python du DAG



Explication



# Étapes importantes du projet

## Consommation des données

Dashboard ou Algorithme de ML

Etape normalement incombant au Data Scientist/Analyst

## Déploiement

Création de l'API de votre projet

Dockerisation de tout le projet



## Remise d'un rapport final du projet.

Rapport expliquant tout le workflow décisionnel du projet

Un Github propre sur lequel le groupe aura travaillé



# Étapes importantes du projet

## Récolte des données

Identification d'une source de données

Connexion et début d'importation des datas

## Architecture & Stockage

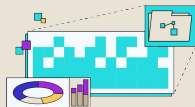
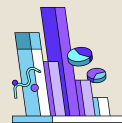
Choix d'un SGBD adapté au problème

Modélisation des tables/collections/index



**Remise d'un document expliquant l'architecture choisie ainsi que la modélisation**





# Logos

