

CryptoWise Copilote

Mehdi FEHRI - Valentin Lagarde - Emile Zeller

2025-01-13

Introduction

Nous allons vous présenter **CryptoWise Copilote**, une application éducative et analytique dédiée au marché des cryptomonnaies.

Notre objectif était de créer une plateforme qui s'adapte au niveau de chaque utilisateur, combinant apprentissage et outils d'analyse avancés sur le marché des cryptos.

Partie 1 : Présentation Fonctionnelle

Notre application est structurée en trois étapes principales :

1. **SmartProfile** : création du profil investisseur
2. Système d'évaluation unique (*des questionnaires basées sur les connaissances financières et cryptos des users*)
3. Les fonctionnalités principales (*CryptoAcademy, Cryptonews, PépitesHunter, Marketscope...*)














I. Création du Profil Investisseur

- ▶ Questionnaire détaillé sur la situation financière du user
- ▶ Évaluation de sa tolérance au risque
- ▶ Définition de ses objectifs d'investissement



















II.Système d'Évaluation Unique

- ▶ Auto-évaluation des ses connaissances
- ▶ 2 Test de compétences de type QCM : *en cryptomonnaies et en finance basés sur 2 banques de questions que nous avons construit et qui inclus + 250 questions tirées aléatoirement*
- ▶ Restitution finale d'un '**Animal type**' et d'un '**Animal Score**' qui détermine l'accès aux fonctionnalités (Pro ou Lite)

Extrait de la Matrice des 27 profils types (*"animal type"*)

capacite	connaissance	risque	animal
Faible	Faible	Élevé	 Dodo
Faible	Faible	Moyen	 Hérisson
Faible	Faible	Faible	 Lapin
Faible	Moyen	Élevé	 Chèvre des falaises
Faible	Moyen	Moyen	 Loutre
Faible	Moyen	Faible	 Canard
Faible	Élevé	Élevé	 Serpent
Faible	Élevé	Moyen	 Hibou
Faible	Élevé	Faible	 Manchot
Moyenne	Faible	Élevé	 Singe hurleur
Moyenne	Faible	Moyen	 blaireau
Moyenne	Faible	Faible	 Paresseux
Moyenne	Moyen	Élevé	 Faucon pèlerin

Matrice des scores obtenus (“animal score”)

Animal	▼	Capacité	▼	Connaissances	▼	Risque	▼	Score	▼↓
 Loup gris		6		6		6		18	
 Lynx		3		6		6		15	
 Ours brun		6		3		6		15	
 Aigle impérial		6		6		3		15	
 Hibou		1		6		6		13	
 Chameau		6		1		6		13	
 Éléphant		6		6		1		13	
 Castor		3		3		6		12	
 Tigre		3		6		3		12	
 Héron		6		3		3		12	
 Loutre		1		3		6		10	
 Serpent		1		6		3		10	
 Blaireau		3		1		6		10	
 Koala		3		6		1		10	
 Guépard		6		1		3		10	
 Tortue géante		6		3		1		10	
 Faucon pèlerin		3		3		3		9	
 Hérisson		1		1		6		8	

III. Les Fonctionnalités Principales

Crypto Academy :

Notre module éducatif comprend 10 chapitres qui englobe tous les aspects des cryptomonnaies, en passant de l'histoire du Bitcoin aux dernières innovations comme la DeFi

Le user peut naviguer librement dans le livre interactif et s'il le souhaite, passer un nouveau test de connaissance

III. Les Fonctionnalités Principales

PépitesHunter : Notre outil d'analyse unique qui mesure les tendances sur les réseaux sociaux et qui combine des données de marché et sentiment des investisseurs

1. Signal prix

- ▶ Identifier les cryptomonnaies ayant connu les plus fortes hausses ou baisses sur 24 heures
- ▶ Fournir une vue d'ensemble rapide des signaux de marché

2. Volume de trading et liquidité

- ▶ Montre l'activité des traders sur un actif
- ▶ Évalue la liquidité des cryptomonnaies

3. Statistiques Sociales

- ▶ Analyse du Sentiment Communautaire

III. Les Fonctionnalités Principales

4. Analyse des Tendances et Recherche Populaire

- ▶ Identifier les projets les plus populaires
- ▶ obtenir des informations détaillées sur les projets
- ▶ Mesure de la Popularité des NFT et des Cryptos les plus recherchés sur Coin Gecko

5. Google Trend

- ▶ Sur la base d'un horizon temporel choisit, présentation des top cryptos les plus recherchées sur le web au niveau mondial

III. Les Fonctionnalités Principales

CryptoNews : Agrégateur d'actualités en temps réel

- ▶ Extraction des 25 dernières dépêches d'actualités liées au monde des cryptomonnaies

III. Les Fonctionnalités Principales

MarketScope : Palette d'outils d'analyse technique et graphique

- ▶ **BitcoinPulse** : Analyse approfondie du Bitcoin (*Crypto de référence*)
- ▶ **AltcoinTracker** : Affiche les dominances du marché des altcoins vs. bitcoin
- ▶ **CryptoFocus** : Analyse détaillée d'une crypto choisie
- ▶ **Market View** : Vue d'ensemble des 100 premières cryptos en terme de capitalisation
- ▶ **Fear & Greed Index** : Indicateur sur l'état psychologique du marché
- ▶ **Trend Scanner** : Détection des tendances émergente
- ▶ **Volume Anomaly** : Détection des anomalies de volume

*“Un aspect unique de notre application est son système de progression. Les utilisateurs commencent avec la **version Lite**, focalisée sur l'apprentissage, et peuvent progresser vers la **version Pro** en démontrant leurs connaissances.”*

Les points forts de notre application sont :

- ▶ Une approche éducative progressive
- ▶ Un système de progression gamifié unique
- ▶ Des outils d'analyse professionnels
- ▶ Une interface intuitive et personnalisée
- ▶ Une intégration de données en temps réel

Immersion dans notre application

Partie 2: Aspects Techniques

- ▶ **7736** lignes de code
- ▶ Plus de **35 fonctions principales**, chacune ayant un rôle spécifique dans le fonctionnement de l'application. Ces fonctions sont organisées de manière modulaire, permettant une maintenance et une évolution faciles du code.

Les fonctions sont réparties en différentes catégories selon leur utilité :

- ▶ Fonctions structurelles de l'application
- ▶ Fonctions d'interface utilisateur
- ▶ Fonctions d'analyse et de traitement des données
- ▶ Fonctions d'affichage et de formatage
- ▶ Fonctions de connexion API
- ▶ Fonctions éducatives et d'évaluation

Voici quelques exemples

1. Une architecture modulaire

```
main <- function() {  
  result1 <- start_app1()  # Profil initial  
  if (!is.null(result1)) {  
    result2 <- start_app2()  # Évaluation  
    if (!is.null(result2)) {  
      start_app3()  # Interface principale  
    }  
  }  
}
```

Cette structure garantit une progression logique et sécurisée de l'utilisateur.

2. Système de Scoring Sophistiqué

```
determine_animal_score <- function(capacite, connaissance,
  # Conversion des critères en points
  capacite_points <- ifelse(capacite == "Élevé", 6,
                           ifelse(capacite == "Moyen", 3, 1))
  connaissance_points <- ifelse(connaissance == "Élevé", 6,
                                ifelse(connaissance == "Moyen", 3, 1))
  risque_points <- ifelse(risque == "Moyen", 6,
                          ifelse(risque == "Élevé", 3, 1))

  # Score total détermine l'accès aux fonctionnalités
  return(capacite_points + connaissance_points + risque_points)
}
```

Ce système permet une évaluation équilibrée des compétences et de la situation de l'utilisateur.

3. Intégration API et Données en Temps Réel

```
get_coingecko_data <- function(endpoint, params = list()) {  
  base_url <- "https://api.coingecko.com/api/v3"  
  url <- paste0(base_url, endpoint)  
  response <- GET(url, query = params)  
  
  if (status_code(response) == 200) {  
    return(fromJSON(rawToChar(response$content)))  
  }  
}
```

Cette fonction est au cœur de nos outils d'analyse, permettant des données de marché en temps réel.

4. Interface Utilisateur Réactive

```
menu_principal <- function(user_info, profil_recap, user_in
  repeat {
    options <- c(
      "Naviguer sur Crypto Academy",
      "Tester vos connaissances",
      "PépitesHunter (PRO)",
      "CryptoNews (PRO)",
      "MarketScope (PRO)"
    )
    # Contrôle d'accès basé sur l'Animal Score
    if (user_info$animal_score >= 13) {
      # Accès complet aux fonctionnalités PRO
    }
  }
}
```

Des menus d'options déroulant qui permettent une meilleure navigation pour le user

5. Les cheats-codes

```
if (choix == "musk") {  
  profil_recap <- mettre_a_jour_profil(profil_recap, user)  
  cat(success_style("\n Comme le roi des cryptotechs, tu  
} else if (choix == "zemmour") {  
  profil_recap <- mettre_a_jour_profil(profil_recap, user)  
  cat(error_style("\n Vous venez d'invoquer Gargamelle..  
}
```

Un aspect ludique qui nous a permis tout de même d'avancer plus rapidement dans le développement du code.

6. Le Système de Style Visuel

```
question_style <- blue$bold  
info_style <- white$italic  
warning_style <- red$bold  
success_style <- green$bold  
highlight_style <- yellow$bold
```

L'interface utilisateur est très travaillée avec un système de couleurs cohérent.

7. Gestion Interactive des Erreurs

```
tryCatch({  
  response <- GET(url, query = params)  
  if (status_code(response) == 429) {  
    cat(warning_style("\nLimite d'API atteinte. Pause c  
    Sys.sleep(61)  
    return(get_coingecko_data(endpoint, params))  
  }  
})
```

La gestion des erreurs d'API avec retry automatique est sophistiquée.

8. Système de Progression Dynamique

```
if (user_info$animal_score < 8) {  
  if (note == 10) {  
    cat(success_style("\nExtraordinaire! Version PRO de  
    profil_recap <- mettre_a_jour_profil(profil_recap,  
  }  
}
```

Le système de progression est bien pensé avec des seuils et récompenses.

9. La gestion des retours en arrière et des sorties de boucles

► Utilisation de repeat avec break

```
repeat {  
  # Exemple dans le profil risque  
  if (state$risk == "0") {  
    current_question <- "age"  
    user_return <- TRUE  
    break # Sort de la boucle repeat  
  }  
}
```

10. Utilisation de next pour revenir au début d'une boucle

```
if (user_return) {  
    next # Retourne au début de la boucle  
}
```

11. Gestion des menus avec return

```
if (choix == "0") {  
    cat(info_style("\nRetour au menu principal...\n"))  
    return()  # Sort complètement de la fonction  
}
```

La structure est hiérarchique

- ▶ break : Sort de la boucle actuelle
- ▶ next : Passe à l'itération suivante
- ▶ return : Sort complètement de la fonction
- ▶ while et repeat : Pour les validations d'entrée

11 Validation des entrées utilisateur

```
while (!confirm %in% c("oui", "non")) {  
  cat(warning_style("Veuillez répondre par 'oui' ou 'non'  
  confirm <- tolower(readline())  
}
```

Conclusion

Merci pour votre attention !

Nous vous invitons à explorer et à vous amuser avec notre code pour découvrir toutes les fonctionnalités de **Cryptowise Copilote**.

Perspectives d'avenir :

- ▶ **Développement de l'application en Shiny** : pour offrir une interface interactive et accessible à tous.
- ▶ **Commercialisation sur l'App Store?**

Explorez, testez et rejoignez-nous dans cette aventure innovante autour des cryptomonnaies.

CryptoWise Copilote est protégé par le droit d'auteur. Toute reproduction ou utilisation sans autorisation préalable est strictement interdite.

