

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE



UNIVERSITE 20 AOUT 1955 SKIKDA

FACULTE DES SCIENCES

DEPARTEMENT D'INFORMATIQUE



Mémoire de fin d'études pour le Master académique en informatique

Spécialité :

Génie Logiciel Avancé et Applications

**Une approche basée sur l'IA pour la gestion
budgétaire**

Réalisé par :

SLIMANE TICH TICH Khaled

Encadré par :

Dr. SEDDARI Nouredine

Année Académique
2023-2024

Reconnaissance et dévouement

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont soutenu et contribué à l'achèvement de ce travail.

Tout d'abord, j'adresse mes sincères remerciements à mon superviseur, Dr SEDDARI Noureddine, pour son accompagnement, son expertise et ses précieux conseils tout au long du processus de recherche. Son soutien constant et ses encouragements ont été essentiels pour mener à bien cette étude.

Je suis également reconnaissant envers les membres du corps professoral de l'Université 20 août, 1955 Skikda pour leur partage de connaissances et leur mentorat. Leur dévouement à l'excellence dans l'éducation et la recherche a créé un environnement propice à mon développement intellectuel.

Ma famille et mes amis méritent également ma reconnaissance pour leur soutien indéfectible, leur compréhension et leurs encouragements tout au long de ce parcours. Leur confiance en moi a été une source constante de motivation.

Enfin, je dédie ce travail à toutes les personnes qui aspirent à la connaissance, à l'innovation et au changement positif. Puissent mes recherches contribuer à faire progresser les sciences informatiques et inspirer d'autres à poursuivre leurs propres aspirations académiques.

Merci sincèrement à tous.

Abstract

The budget management application represents a significant advancement in financial technology, integrating artificial intelligence to provide users with advanced tools for financial planning and management. This end-of-study memoir project explores how the application utilizes AI to offer personalized recommendations based on user behaviors, preferences, and financial data, thereby facilitating budget management, expense tracking, and financial goal setting.

Designed to be user-friendly, the application features intuitive interfaces and interactive visualizations such as dynamic graphs, allowing users to easily track expenses, monitor income, and analyze financial trends.

This memoir project also highlights the application's continuous evolution through regular updates based on user feedback and AI analysis. Looking forward, it aims to enhance AI predictive capabilities, further automate expense management, and integrate new technologies to optimize financial management.

Résumé

L'application de gestion budgétaire représente une avancée significative dans le domaine de la technologie financière, intégrant l'intelligence artificielle pour fournir aux utilisateurs des outils avancés de planification et de gestion financière. Ce projet de mémoire explore comment l'application utilise l'IA pour offrir des recommandations personnalisées basées sur les comportements, préférences et données financières des utilisateurs, facilitant ainsi la gestion du budget, des dépenses et l'établissement d'objectifs financiers.

Conçue pour être conviviale, l'application propose des interfaces intuitives et des visualisations interactives telles que des graphiques dynamiques, permettant aux utilisateurs de suivre facilement leurs dépenses, surveiller leurs revenus et analyser les tendances financières.

Ce projet de mémoire met également en lumière l'évolution continue de l'application à travers des mises à jour régulières basées sur les retours des utilisateurs et les analyses AI. Pour l'avenir, il envisage d'approfondir les capacités prédictives de l'IA, d'automatiser davantage la gestion des dépenses et d'intégrer de nouvelles technologies pour optimiser la gestion financière.

الملخص

تمثل تطبيق إدارة الميزانية تقدمًا كبيرًا في التكنولوجيا المالية، حيث يدمج الذكاء الاصطناعي لتزويد المستخدمين بأدوات متقدمة للتخطيط والإدارة المالية. يستكشف هذا المشروع لمذكرة نهاية الدراسة كيفية استخدام التطبيق للذكاء الاصطناعي لتقديم توصيات مخصصة استنادًا إلى سلوكيات المستخدمين وتفضيلاتهم والبيانات المالية، مما يسهل إدارة الميزانية وتتبع النفقات وتحديد الأهداف المالية.

صمم التطبيق ليكون سهل الاستخدام، حيث يتميز بواجهات بديهية وتصورات تفاعلية مثل الرسوم البيانية الديناميكية، مما يتيح للمستخدمين تتبع النفقات بسهولة ومراقبة الدخل وتحليل الاتجاهات المالية.

يسلط هذا المشروع لمذكرة نهاية الدراسة الضوء أيضًا على التطور المستمر للتطبيق من خلال التحديثات المنتظمة استنادًا إلى ملاحظات المستخدمين وتحليلات الذكاء الاصطناعي. يهدف التطبيق في المستقبل إلى تعزيز قدرات التنبؤ بالذكاء الاصطناعي، وتأمين إدارة النفقات بشكل أكبر، ودمج التقنيات الجديدة لتحسين الإدارة المالية.

في الختام، يوضح هذا المشروع لمذكرة نهاية الدراسة كيفية إعادة تعريف تطبيق إدارة الميزانية للإدارة المالية الحديثة من خلال تقديم أدوات شخصية ومنتكيفة تعزز الكفاءة والتمكين في التخطيط المالي، مع استكشاف الآفاق المستقبلية للتطوير والابتكار في هذا المجال.

Table des matières

Introduction générale :	1
Chapitre 1 Introduction aux applications mobiles de gestion budgétaire	3
1.1 Introduction	4
1.2 Introduction aux applications mobiles de gestion budgétaire	4
1.2.1 Contexte et évolution des applications mobiles de gestion budgétaire	4
1.2.2 Importance croissante des technologies mobiles dans la gestion budgétaire	5
1.2.3 Impact sur les comportements financiers et la prise de décision	6
1.3 Fonctionnalités essentielles des applications mobiles de gestion budgétaire	8
1.3.1 Suivi des finances personnelles	8
1.3.2 Création et gestion de budgets	8
1.3.3 Planification financière avancée	8
1.3.4 Analyse des tendances et des habitudes de dépenses	9
1.3.5 Alertes et notifications personnalisées	9
1.3.6 Sécurité et protection des données	9
1.3.7 Intégration avec d'autres services financiers	9
1.4 Innovations récentes et tendances dans les applications mobiles de gestion budgétaire	10
1.4.1 Intelligence artificielle et machine learning	10
1.4.2 Automatisation des finances personnelles	11
1.4.3 Personnalisation et recommandations contextuelles	12
1.4.4 Sécurité renforcée et confidentialité des données	13
1.5 Défis et opportunités dans le développement et l'adoption des applications mobiles de gestion budgétaire	14
1.5.1 Défis	14
1.5.2 Opportunités	15
1.6 Conclusion	15
Chapitre 2 : Machine learning	16
2.1. Introduction	17
2.2 Définition	17
2.3 Types de Machine learning	18

2.3.1 Apprentissage supervisé	18
2.3.2 Apprentissage non supervisé	19
2.3.3 Apprentissage par renforcement	19
2.3.4 Apprentissage semi-supervisé.....	20
2.4 Algorithmes de Machine Learning	20
2.4.1. Decision Tree (arbre de décision).....	20
2.4.2. Random Forest.....	23
2.4.3. Régression linéaire et logistique (Logistic regression).....	25
2.4.4. Support Vector Machine (SVM)	26
2.4.5. KNN (k-Nearest-Neighbors).....	28
2.4.6 Réseaux de neurones artificiels.....	29
2.5. Evaluation et validation	31
2.6. Mesure de performance	31
2.6.1. Confusion Matrix (matrice de confusion).....	31
2.6.2. Métriques d'évaluation	32
2.6.3. Courbes d'évaluation-Receiver Operating Characteristics Curve (ROC Curve).....	34
2.6.4. Aire sous la courbe ROC (Area Under Curve - AUC)	36
2.6.5. Cross Validation (CV).....	37
Chapitre 3: Machine learning dans le domaine des applications mobiles pour la gestion budgétaire.....	39
3.1 Introduction.....	40
3.2 Qu'est-ce que l'apprentissage automatique et comment s'applique-t-il à la gestion budgétaire ?	40
3.3 Comment l'apprentissage supervisé peut-il améliorer les prédictions financières ?	41
3.4 Quelles perspectives l'apprentissage non supervisé peut-il offrir dans le domaine de la gestion du budget ?.....	42
3.5 Comment l'apprentissage par renforcement peut-il optimiser la gestion du budget ?	44
3.6 Quels sont les défis et limites de l'application de l'apprentissage machine dans la gestion du budget ?	45
3.7 Conclusion	47
Chapitre 4 méthodologie	48
4.1 Introduction à la Méthodologie.....	49

4.2 Langage de Modélisation Unifié (UML) et Ses Diagrammes	49
4.2.1 Comprendre l'UML.....	49
4.2.2 Aperçu des Diagrammes UML.....	49
4.3 Diagrammes UML clés dans le développement de l'application budget app	50
4.4 c'est quoi budget app ?	51
4.4.1 Budget App fonctionnalité	51
4.4.2 Diagramme de cas d'utilisation	55
4.4.3 Diagramme de séquence	62
4.4.4 Diagramme de classe	69
4.5 Modèle Proposé	70
4.5.1 Introduction	70
4.5.2 Collecte et prétraitement des données.....	70
4.5.4 Sélection de fonctionnalité	71
4.5.5 Architecture du modèle	71
4.5.6 Processus de formation	72
4.5.7 Paramètres d'évaluation	73
4.5.8 Conversion du modèle	73
4.5.9 Déploiement et tests.....	74
4.6 Conclusion	75
Chapitre 5 : implémentation	76
5.1 Introduction.....	77
5.2 Architecture Model View ViewModel (MVVM)	77
5.2.1 Modèle (Model).....	77
5.2.2 Vue (View).....	77
5.2.3 Vue Modèle (ViewModel)	77
5.2.4 Interaction entre les composants	78
5.2.5 Avantages de l'Architecture MVVM.....	78
5.3 Matériel utilisé.....	78
5.4 Technologie et Framework	79
5.4.1 Langage de programmation	79
5.4.2 Frameworks et bibliothèques	79

5.4.3 Outils de développement	81
5.5 Implémentation du Modèle.....	82
5.6 Interface utilisateur	85
5.7 Test.....	93
5.8 Résultat statistique.....	94
5.9 Conclusion	95
Conclusion Générale	96
Les Références	98

Figures

- Figure 1 : Arbre de décision pour classification fit/unfit 21
- Figure 2 : Exemple illustrant le déroulement de Random Forest 24
- Figure 3 : Variations de la fonction logistique 26
- Figure 4 : Fonctionnement du modèle Support Vector Machine 27
- Figure 5 : Exemple de classification KNN (K=3 et K=5) 29
- Figure 6 : Visualisation (a) des neurones biologiques et artificiels et (b) un exemple d'un réseau neuronal..... 30
- Figure 7 : Confusion Matrix 31
- Figure 8 : Courbe ROC 35
- Figure 9 : AUC 36
- Figure 10 : Diagramme de cas d'utilisation de la conception Budget App 55
- Figure 11 : Diagramme de séquence représentant la phase d'inscription 62
- Figure 12 : Diagramme de séquence représentant la page d'accueil 63
- Figure 13 : Diagramme de séquence représentant le Button IA..... 64
- Figure 14 : Diagramme de séquence représentant la page Graphique 65
- Figure 15 : Diagramme de séquence représentant la page Dettes/Crédits 66
- Figure 16 : Diagramme de séquence représentant les Budgets 67
- Figure 17 : Diagramme de séquence représentant les objectifs 68
- Figure 18 : Diagramme de classe de la conception Budget App 69
- Figure 19 : Collecte et prétraitement des données 70
- Figure 20 : Feature importance and importance..... 71
- Figure 21 : Architecture du modèle..... 72
- Figure 22 : train_loss and val_loss..... 72
- Figure 23 : Mean absolute error/ squared error 73
- Figure 24 : conversion du modèle 74
- Figure 25 : déploiement et tests 74
- Figure 26 : code pour la préparation des données..... 82
- Figure 27 : code pour model architecture 83
- Figure 28 : code pour l'entraînement 83
- Figure 29 : code pour recommandation personnalisées 84
- Figure 30 : code pour conversion en tensorflow lite 84
- Figure 31 : page de connexion pour l'application Budget App 85
- Figure 32 : Menu de l'application Budget App 86
- Figure 33 : page d'Accueil pour l'application Budget App 87
- Figure 34 : page d'ajoute dépense/revenu pour l'application Budget App 88
- Figure 35 : page Budgets pour l'application Budget App 89

Figure 36 : page dettes/crédits pour l'application Budget App	90
Figure 37 : page Graphique pour l'application Budget App.....	91
Figure 38 : génération des objectifs pour Budget App	92
Figure 39 : statistique entre train_loss et val_loss pour le model de recommandation.....	95

Introduction générale :

L'intelligence artificielle (IA) a le potentiel de révolutionner divers secteurs et de transformer notre quotidien. En particulier, le domaine de l'apprentissage automatique offre des perspectives prometteuses dans la gestion budgétaire. Il permet le développement de solutions personnalisées et adaptables qui peuvent efficacement aider les individus à gérer leurs finances personnelles.

À l'échelle mondiale, les problèmes financiers comme l'endettement, le manque d'épargne et la mauvaise gestion des dépenses deviennent de plus en plus préoccupants. Ces difficultés découlent souvent de mauvaises habitudes financières et d'une méconnaissance des principes de gestion financière, posant ainsi des défis significatifs tant pour les individus que pour les systèmes financiers globaux. Il est donc essentiel de mettre en place des solutions efficaces pour promouvoir des comportements financiers plus sains, mettant l'accent sur la gestion budgétaire et l'épargne.

Ces dernières années, l'industrie des technologies financières a été profondément transformée par l'émergence d'applications mobiles dédiées à la gestion budgétaire. Ces applications se sont avérées être des outils prometteurs pour encourager une gestion financière régulière, améliorer les niveaux d'épargne et optimiser les résultats financiers. Elles offrent une gamme de fonctionnalités allant du suivi des dépenses à la création de budgets personnalisés, en passant par le suivi des progrès vers les objectifs financiers.

Cependant, une limitation majeure de nombreuses applications actuelles de gestion budgétaire réside dans leur approche uniforme. Elles proposent souvent des conseils et des budgets génériques qui ne prennent pas en compte les besoins, préférences et contextes uniques des utilisateurs. Par exemple, un professionnel disposant d'un revenu stable pourrait avoir besoin de conseils en matière d'investissement et d'épargne, tandis qu'un étudiant avec un budget limité pourrait privilégier des conseils sur la gestion quotidienne des dépenses et l'évitement des dettes.

De nombreuses études soulignent l'importance de la personnalisation dans les interventions financières. Elles démontrent que des plans de gestion financière personnalisés entraînent des améliorations significatives des comportements financiers par rapport aux conseils standard. Ainsi, il devient de plus en plus crucial de développer des applications de gestion budgétaire capables de fournir des recommandations financières personnalisées. Ces applications doivent intégrer divers facteurs tels que les revenus et sources de revenus, les dépenses et leur catégorisation, les objectifs financiers, ainsi que les habitudes et comportements financiers des utilisateurs, afin d'offrir des conseils budgétaires pertinents et adaptés. Cette approche permettrait non seulement d'améliorer l'expérience utilisateur, mais aussi potentiellement d'accroître l'adhésion aux bonnes pratiques de gestion financière et d'optimiser les résultats financiers globaux.

Cette étude vise à approfondir la compréhension du concept d'apprentissage automatique et de son application dans la gestion budgétaire.

Dans le cadre de cette étude, une application de gestion budgétaire baptisée "budget app" a été développée. Budget app utilise le Machine Learning pour proposer des gestions financières personnalisés budgétaires en fonction de chaque utilisateur, tel que son état financier actuel. L'étude discute également de l'impact potentiel de Budget app sur l'engagement des utilisateurs, leur adhésion aux bonnes pratiques budgétaires et les résultats financiers globaux, en identifiant les futures orientations de recherche dans l'application du Machine learning aux systèmes de recommandation budgétaire.

Les conclusions de cette étude pourraient fournir des informations précieuses aux développeurs d'applications mobiles de gestion budgétaire, en leur offrant les connaissances nécessaires pour concevoir des programmes financiers plus personnalisés et efficaces pour leurs utilisateurs.

Chapitre 1 Introduction aux applications mobiles de gestion budgétaire

1.1 Introduction

Les applications mobiles de gestion budgétaire représentent une réponse contemporaine aux besoins croissants de gestion financière personnelle et professionnelle dans un monde de plus en plus connecté. Cette section explore en profondeur l'émergence et l'évolution de ces applications, leur importance croissante, ainsi que leur impact sur la manière dont les individus et les entreprises gèrent leurs finances au quotidien.

1.2 Introduction aux applications mobiles de gestion budgétaire

1.2.1 Contexte et évolution des applications mobiles de gestion budgétaire

Les applications mobiles de gestion budgétaire ont évolué de manière significative au fil des années, influencées par l'avancement rapide des technologies mobiles et les changements dans les comportements des consommateurs en matière de gestion financière. Initialement, ces applications étaient essentiellement des outils simples permettant aux utilisateurs de suivre leurs dépenses et de gérer leurs budgets sur des appareils mobiles. Leur développement a été stimulé par plusieurs facteurs clés :

- **Avènement des smartphones** : L'introduction et la popularisation des smartphones au début des années 2000 ont ouvert de nouvelles possibilités pour le développement d'applications mobiles plus sophistiquées. Les capacités croissantes de ces appareils en termes de puissance de traitement, de mémoire et de connectivité ont permis aux développeurs de créer des applications plus complexes et interactives pour la gestion financière.
- **Intégration avec d'autres technologies** : L'intégration des applications de gestion budgétaire avec d'autres technologies, telles que les services bancaires en ligne, les plateformes de paiement mobile et les agrégateurs de finances personnelles, a également

joué un rôle crucial dans leur évolution. Cette interconnectivité permet aux utilisateurs de centraliser leurs informations financières et d'automatiser certaines tâches, comme la catégorisation des dépenses ou la planification budgétaire.

- **Innovation continue** : Les développeurs continuent d'innover dans le domaine des applications mobiles de gestion budgétaire, en introduisant des fonctionnalités avancées telles que l'intelligence artificielle pour l'analyse prédictive des dépenses, les assistants virtuels pour des conseils financiers personnalisés, et la sécurité renforcée pour protéger les informations sensibles des utilisateurs.

1.2.2 Importance croissante des technologies mobiles dans la gestion budgétaire

L'importance des technologies mobiles dans la gestion budgétaire moderne ne peut être sous-estimée, car elles offrent des avantages significatifs qui facilitent la gestion financière personnelle et professionnelle de manière proactive et efficace. Cette section explore en détail pourquoi les technologies mobiles sont devenues indispensables dans ce domaine [1].

Accessibilité et disponibilité constante : Les smartphones sont devenus des compagnons omniprésents dans la vie quotidienne, permettant aux utilisateurs d'accéder instantanément à leurs applications de gestion budgétaire où qu'ils soient et à tout moment. Cette accessibilité 24/7 est cruciale pour une gestion proactive des finances, car elle permet aux utilisateurs de suivre en temps réel leurs transactions, de vérifier leurs soldes et de faire des ajustements immédiats à leurs budgets.

Simplicité et facilité d'utilisation : Les applications mobiles de gestion budgétaire sont conçues pour être intuitives et conviviales, même pour les utilisateurs sans expertise financière approfondie. Elles utilisent des interfaces simples et des graphiques interactifs pour rendre la gestion financière plus accessible et compréhensible, ce qui encourage une utilisation régulière et une meilleure adhésion aux pratiques de gestion budgétaire.

Automatisation des tâches fastidieuses : Les technologies mobiles permettent l'automatisation de nombreuses tâches liées à la gestion budgétaire, telles que la catégorisation automatique des

dépenses, la planification budgétaire récurrente et l'alerte en cas de dépassement de budget. Cela libère du temps pour les utilisateurs et réduit les erreurs humaines, améliorant ainsi l'efficacité globale de la gestion financière.

Intégration avec d'autres services financiers : Les applications mobiles de gestion budgétaire sont souvent intégrées avec d'autres services financiers, tels que les banques en ligne, les agrégateurs de comptes et les plateformes de paiement mobile. Cette interconnectivité permet aux utilisateurs de centraliser toutes leurs informations financières en un seul endroit, facilitant la visualisation globale de leur situation financière et simplifiant les processus de gestion.

Personnalisation et recommandations intelligentes : Grâce à l'utilisation de l'intelligence artificielle et du machine learning, certaines applications mobiles sont capables de fournir des recommandations personnalisées basées sur les habitudes de dépenses des utilisateurs, les objectifs financiers définis et les tendances économiques. Ces fonctionnalités aident les utilisateurs à prendre des décisions plus éclairées et à optimiser leur gestion financière pour atteindre leurs objectifs à long terme.

Sécurité des données renforcée : Avec des préoccupations croissantes concernant la sécurité des données, les développeurs d'applications mobiles de gestion budgétaire mettent en œuvre des mesures robustes pour protéger les informations financières sensibles des utilisateurs. Cela inclut l'utilisation de cryptage avancé, de méthodes d'authentification sécurisées et de pratiques de sécurité conformes aux normes industrielles.

1.2.3 Impact sur les comportements financiers et la prise de décision

Les applications mobiles de gestion budgétaire ont profondément transformé les comportements financiers des utilisateurs en facilitant une gestion proactive et informée de leurs finances personnelles et professionnelles. Leur impact se manifeste à plusieurs niveaux, influençant directement la manière dont les individus prennent des décisions financières et gèrent leurs ressources [2].

Visibilité et transparence accrues : Les applications mobiles permettent aux utilisateurs d'avoir une vision claire et instantanée de leur situation financière. Grâce à des interfaces conviviales et

à des tableaux de bord interactifs, les utilisateurs peuvent visualiser leurs dépenses, leurs revenus, leurs soldes de compte et leurs objectifs financiers en temps réel. Cette transparence renforce la responsabilité financière et encourage une surveillance continue des habitudes de dépenses.

Suivi et analyse des habitudes de dépenses : En capturant et en catégorisant automatiquement les transactions, les applications mobiles facilitent le suivi détaillé des habitudes de dépenses. Les utilisateurs peuvent identifier facilement les domaines où l'argent est dépensé, ce qui leur permet de détecter les gaspillages potentiels, d'ajuster leurs budgets et de prendre des décisions éclairées pour optimiser leur utilisation des fonds disponibles.

Planification financière proactive : Avec des outils intégrés pour créer et suivre des budgets personnalisés, les utilisateurs peuvent planifier leurs finances à l'avance et anticiper les flux de trésorerie futurs. Les fonctionnalités de planification incluent souvent des alertes en cas de dépassement de budget, des prévisions de revenus et des simulateurs pour explorer différents scénarios financiers. Cette approche proactive aide les utilisateurs à éviter les difficultés financières imprévues et à atteindre leurs objectifs financiers à long terme.

Décisions informées et optimisées : Les analyses avancées fournies par certaines applications mobiles utilisant l'intelligence artificielle permettent aux utilisateurs de recevoir des recommandations personnalisées basées sur leurs données financières et leurs objectifs. Par exemple, ces applications peuvent suggérer des ajustements de budget, des stratégies d'épargne optimisées ou des opportunités d'investissement alignées avec les préférences et les tolérances au risque de l'utilisateur. Cette capacité à prendre des décisions informées améliore significativement la gestion financière globale.

Éducation financière continue : En fournissant des conseils financiers personnalisés et des ressources éducatives intégrées, les applications mobiles jouent un rôle crucial dans l'amélioration de l'éducation financière des utilisateurs. Ils peuvent apprendre les principes de base de la gestion budgétaire, la planification financière à long terme, et les meilleures pratiques pour économiser et investir efficacement.

Réduction du stress financier : En automatisant les tâches de gestion financière et en offrant des outils de suivi et d'optimisation, les applications mobiles contribuent à réduire le stress lié à la gestion quotidienne des finances. Les utilisateurs se sentent plus en contrôle de leur argent, ce qui peut améliorer leur bien-être général et réduire les tensions financières au sein des ménages et des petites entreprises.

1.3 Fonctionnalités essentielles des applications mobiles de gestion budgétaire

Les applications mobiles de gestion budgétaire sont conçues pour offrir une gamme diversifiée de fonctionnalités visant à simplifier la gestion financière personnelle et professionnelle

1.3.1 Suivi des finances personnelles

Les applications permettent aux utilisateurs de capturer facilement et de catégoriser automatiquement leurs transactions financières à partir de diverses sources telles que les comptes bancaires, les cartes de crédit et les portefeuilles électroniques. Ce suivi détaillé fournit une vue d'ensemble instantanée des dépenses et des revenus, facilitant ainsi une gestion plus précise des finances quotidiennes [3].

1.3.2 Création et gestion de budgets

Les fonctionnalités de budget permettent aux utilisateurs de définir des objectifs financiers, d'allouer des montants spécifiques à différentes catégories de dépenses et de suivre la progression par rapport à ces budgets. Les applications peuvent générer des rapports et des graphiques pour visualiser la conformité aux budgets établis et identifier les domaines où des ajustements sont nécessaires.

1.3.3 Planification financière avancée

Certaines applications offrent des outils de planification financière qui aident les utilisateurs à prévoir et à gérer les revenus futurs, les dépenses récurrentes, les paiements de prêts, ainsi que l'épargne et les investissements. Ces fonctionnalités incluent souvent des calendriers financiers,

des simulateurs de prêt, et des calculateurs d'intérêt pour faciliter une gestion proactive des finances à long terme.

1.3.4 Analyse des tendances et des habitudes de dépenses

À l'aide de l'intelligence artificielle et du machine learning, certaines applications analysent les habitudes de dépenses des utilisateurs pour identifier les tendances, les motifs de dépenses et les opportunités d'optimisation. Ces analyses permettent aux utilisateurs de mieux comprendre où va leur argent et de prendre des décisions informées pour ajuster leurs comportements financiers [4].

1.3.5 Alertes et notifications personnalisées

Les notifications en temps réel avertissent les utilisateurs des transactions importantes, des paiements à venir, des dépassements de budget ou des tendances anormales dans les dépenses. Ces alertes aident à maintenir une surveillance constante sur les finances et à réagir rapidement aux événements financiers critiques.

1.3.6 Sécurité et protection des données

La sécurité des données est une priorité pour les applications de gestion budgétaire. Elles utilisent des mesures de sécurité avancées telles que le cryptage des données, l'authentification multi-facteurs et la conformité aux normes de sécurité financière pour protéger les informations sensibles des utilisateurs contre les cybermenaces et les accès non autorisés.

1.3.7 Intégration avec d'autres services financiers

Pour une gestion financière plus intégrée, les applications peuvent se connecter directement aux comptes bancaires, aux cartes de crédit et aux autres services financiers. Cela permet une synchronisation automatique des transactions et des soldes, éliminant ainsi le besoin de saisir manuellement les données et assurant une mise à jour en temps réel des informations financières.

1.4 Innovations récentes et tendances dans les applications mobiles de gestion budgétaire

1.4.1 Intelligence artificielle et machine learning

L'intelligence artificielle (IA) et le machine learning (ML) jouent un rôle croissant dans l'évolution des applications mobiles de gestion budgétaire, offrant des capacités avancées pour améliorer l'efficacité, la personnalisation et l'analyse des données financières des utilisateurs. Voici comment ces technologies sont intégrées et utilisées [6]

Analyse prédictive des dépenses : Les algorithmes d'IA et de ML analysent les habitudes de dépenses passées des utilisateurs pour prédire leurs dépenses futures. En utilisant des modèles statistiques et des données historiques, ces applications peuvent estimer avec précision les montants susceptibles d'être dépensés dans différentes catégories, aidant ainsi les utilisateurs à planifier leur budget plus efficacement .

Recommandations budgétaires personnalisées : En analysant les données financières et les objectifs définis par l'utilisateur, les systèmes d'IA peuvent générer des recommandations personnalisées pour optimiser le budget. Par exemple, ils peuvent suggérer des ajustements de budget basés sur les habitudes de dépenses récentes, les changements dans les revenus, ou les événements de vie importants comme l'achat d'une maison ou la naissance d'un enfant.

Détection des anomalies et des fraudes : Les techniques d'apprentissage automatique sont utilisées pour détecter les transactions suspectes ou non autorisées en comparant les comportements de dépenses habituels des utilisateurs avec les modèles attendus. Cela permet aux applications de signaler rapidement toute activité potentiellement frauduleuse, renforçant ainsi la sécurité des comptes financiers des utilisateurs.

Assistance virtuelle et chatbots : Certains applications intègrent des assistants virtuels alimentés par l'IA pour fournir un support client personnalisé et des réponses aux questions financières courantes. Ces chatbots peuvent aider les utilisateurs à comprendre leurs finances, à naviguer dans l'application, et à recevoir des conseils en temps réel, améliorant ainsi l'expérience utilisateur globale.

Personnalisation de l'expérience utilisateur : Grâce à l'apprentissage automatique, les applications peuvent adapter leur interface et leurs recommandations en fonction des interactions passées de l'utilisateur, de ses préférences et de ses besoins spécifiques. Par exemple, elles peuvent proposer des fonctionnalités pertinentes en fonction de la situation financière actuelle de l'utilisateur ou des événements de vie récents.

Analyse avancée des données financières : Les algorithmes d'IA permettent une analyse approfondie des données financières, facilitant la génération de rapports détaillés, la prévision des tendances économiques et la simulation de scénarios financiers. Ces analyses aident les utilisateurs à prendre des décisions éclairées basées sur des informations précises et actualisées.

1.4.2 Automatisation des finances personnelles

L'automatisation des finances personnelles est une fonctionnalité clé des applications mobiles de gestion budgétaire, visant à simplifier la gestion quotidienne des finances des utilisateurs en automatisant divers processus [6].

Paiement de factures automatique : Les applications permettent aux utilisateurs de configurer des paiements automatiques pour les factures récurrentes telles que les services publics, l'assurance, le loyer, etc. Une fois configurée, cette fonctionnalité garantit que les paiements sont effectués à temps chaque mois, évitant ainsi les frais de retard et assurant une bonne gestion des dépenses fixes.

Épargne automatique : Cette fonctionnalité permet aux utilisateurs de définir des règles pour transférer automatiquement une partie de leurs revenus vers des comptes d'épargne ou des fonds d'investissement. Par exemple, ils peuvent choisir de mettre de côté un pourcentage fixe de leur salaire chaque mois ou de transférer automatiquement un montant spécifique après chaque dépôt de paie. Cela facilite l'accumulation d'économies sans effort supplémentaire.

Gestion intelligente des remboursements de prêts : Les applications peuvent suivre les prêts en cours et automatiser les paiements de prêt, en veillant à ce que les utilisateurs respectent leurs échéances mensuelles. Elles peuvent également calculer automatiquement les intérêts et

les amortissements, offrant ainsi une vue d'ensemble claire de l'avancement du remboursement du prêt.

Optimisation des budgets : En utilisant des algorithmes d'intelligence artificielle, certaines applications peuvent analyser les habitudes de dépenses des utilisateurs et ajuster automatiquement leurs budgets en fonction des changements dans les revenus ou les dépenses. Par exemple, elles peuvent réallouer les montants alloués à différentes catégories de dépenses en fonction des tendances observées, optimisant ainsi l'utilisation des ressources financières.

1.4.3 Personnalisation et recommandations contextuelles

La personnalisation et les recommandations contextuelles sont des aspects essentiels des applications mobiles de gestion budgétaire modernes, visant à adapter l'expérience utilisateur en fonction des besoins individuels et des situations financières spécifiques [6].

Analyse des habitudes de dépenses : Les applications analysent les habitudes de dépenses passées des utilisateurs pour comprendre leurs comportements financiers uniques. Cela inclut l'identification des catégories de dépenses prédominantes, des périodes de pic de dépenses, et des tendances saisonnières ou annuelles qui influencent les finances personnelles.

Recommandations personnalisées : En utilisant des algorithmes d'intelligence artificielle et de machine learning, les applications génèrent des recommandations personnalisées pour optimiser le budget et améliorer la gestion financière. Par exemple, elles peuvent suggérer des ajustements de budget basés sur les habitudes de dépenses récentes, les changements dans les revenus, ou des événements de vie tels que des mariages ou des naissances.

Utilisation de données contextuelles : Les applications intègrent des données contextuelles telles que la localisation géographique, les événements de vie importants, et les conditions économiques locales pour fournir des recommandations pertinentes. Par exemple, elles peuvent suggérer des économies supplémentaires avant un voyage ou ajuster le budget après une augmentation salariale, en tenant compte du contexte spécifique de l'utilisateur.

Notifications personnalisées : Les utilisateurs reçoivent des notifications en temps réel basées sur leur comportement financier et leurs objectifs définis. Ces notifications peuvent inclure des rappels de paiement, des alertes de dépenses excessives, ou des conseils pour maximiser les économies, ce qui aide à maintenir une gestion proactive des finances personnelles.

Analyse comparative et benchmarking : Certaines applications offrent des fonctionnalités de benchmarking qui permettent aux utilisateurs de comparer leurs habitudes de dépenses et leurs performances financières avec d'autres utilisateurs similaires. Cela peut fournir des perspectives utiles sur les domaines où des améliorations peuvent être apportées ou des opportunités d'optimisation financière.

Évolution adaptative : Les systèmes d'IA et de ML sont conçus pour apprendre et s'adapter en continu en fonction des nouvelles données et des interactions des utilisateurs. Cela garantit que les recommandations et les conseils financiers restent pertinents et efficaces au fil du temps, en tenant compte des changements dans la vie financière et personnelle de l'utilisateur.

1.4.4 Sécurité renforcée et confidentialité des données

La sécurité renforcée et la confidentialité des données sont des aspects cruciaux des applications mobiles de gestion budgétaire, étant donné la nature sensible des informations financières qu'elles traitent.

Chiffrement des données au repos : Les informations stockées sur les serveurs de l'application sont également chiffrées à l'aide de techniques de chiffrement avancées (par exemple, AES-256). Cela garantit que même si les données sont compromises, elles restent illisibles sans les clés de déchiffrement appropriées.

Renforcement de l'authentification : La 2FA ajoute une couche supplémentaire de sécurité en exigeant que les utilisateurs fournissent deux formes de vérification avant d'accéder à leur compte. Cela peut inclure un mot de passe et un code envoyé à leur téléphone ou un appareil de sécurité comme une clé USB. Même si un pirate informatique parvient à obtenir le mot de passe de l'utilisateur, il ne pourra pas accéder au compte sans le deuxième facteur de vérification.

Accès limité basé sur les rôles : Les applications implémentent des contrôles d'accès stricts pour garantir que seuls les utilisateurs autorisés puissent accéder aux informations sensibles. Les employés de l'entreprise qui gère l'application n'ont accès qu'aux données nécessaires pour accomplir leurs tâches spécifiques.

1.5 Défis et opportunités dans le développement et l'adoption des applications mobiles de gestion budgétaire

Les applications mobiles de gestion budgétaire ont révolutionné la manière dont les individus gèrent leurs finances personnelles, mais leur développement et adoption ne sont pas sans défis.

1.5.1 Défis

Sécurité des données et confidentialité : Les applications doivent constamment évoluer pour se défendre contre des menaces de sécurité de plus en plus sophistiquées, ce qui nécessite des investissements dans des technologies de sécurité avancées et des audits réguliers pour détecter et corriger les vulnérabilités.

Intégration avec diverses institutions financières : Les applications doivent être compatibles avec une large gamme de banques, de cartes de crédit et d'autres institutions financières, mais l'absence de normes uniformes pour l'intégration des API complique ce processus, rendant l'intégration technique plus complexe et coûteuse. Obtenir un accès en temps réel aux données financières des utilisateurs peut être difficile, car toutes les institutions financières ne fournissent pas des API robustes et actualisées.

Complexité de l'analyse des données : Les applications utilisent des algorithmes d'IA et de ML pour fournir des recommandations et des analyses financières, mais assurer l'exactitude et la fiabilité de ces algorithmes est un défi nécessitant des volumes de données importants et des techniques de validation rigoureuses. La gestion et l'analyse de grandes quantités de données financières en temps réel nécessitent des infrastructures robustes et des capacités de calcul avancées.

1.5.2 Opportunités

Amélioration de la santé financière personnelle : Les applications peuvent jouer un rôle clé dans l'amélioration de l'éducation financière des utilisateurs en leur fournissant des outils et des ressources pour mieux comprendre et gérer leurs finances. Grâce à l'IA et au ML, elles peuvent offrir des recommandations personnalisées qui aident les utilisateurs à optimiser leurs budgets, à épargner davantage et à atteindre leurs objectifs financiers.

Intégration de technologies émergentes : L'intégration de la blockchain et des cryptomonnaies peut offrir des solutions de paiement sécurisées, des contrats intelligents et des services d'investissement innovants. L'utilisation d'assistants virtuels et de chatbots alimentés par l'IA peut améliorer l'engagement des utilisateurs et fournir un support client efficace et personnalisé.

Analyse prédictive et intelligence financière : Les algorithmes de ML peuvent prédire les tendances de dépenses et de revenus, permettant aux utilisateurs de planifier plus efficacement et de prendre des décisions financières éclairées. Les applications peuvent également offrir des conseils d'investissement basés sur l'analyse des marchés financiers et les objectifs personnels des utilisateurs, améliorant ainsi le rendement des investissements.

1.6 Conclusion

Les applications mobiles de gestion budgétaire ont émergé comme des outils indispensables pour la gestion des finances personnelles à l'ère numérique. L'introduction de ces applications a transformé la manière dont les individus surveillent, planifient et contrôlent leurs dépenses, en offrant une accessibilité et une commodité sans précédent. Les fonctionnalités essentielles de ces applications, telles que le suivi des dépenses, l'analyse budgétaire, les alertes personnalisées et les recommandations financières, ont permis aux utilisateurs de mieux comprendre et optimiser leur situation financière.

Chapitre 2 : Machine learning

2.1. Introduction

Arthur Samuel un informaticien d'IBM a écrit un programme informatique pour jouer aux dames en 1959. Même si les règles qui régissent le jeu sont assez peu complexes, il faut savoir qu'il existe des milliards de situations possibles et il est par conséquent quasiment impossible de programmer explicitement l'ordinateur avec des instructions sur ce qu'il faut faire dans chaque situation. Au début, les scores étaient basés sur une formule utilisant des facteurs tels que le nombre de pièces de chaque côté et le nombre de rois. Cette approche fonctionnait, cependant Samuel avait déjà une vision de comment améliorer ses performances. Il a fait jouer le programme des milliers de parties contre lui-même et a utilisé les résultats (victoire ou défaite) pour évaluer la probabilité de gagner en effectuant un mouvement donné. Arthur avait donc écrit un programme informatique capable d'améliorer ses propres performances grâce à l'expérience et c'est ainsi que l'apprentissage automatique a vu le jour.

2.2 Définition

L'apprentissage automatique (machine learning) est une branche de l'intelligence artificielle (I.A) et un domaine d'étude qui vise à donner aux ordinateurs la capacité d'apprendre et d'améliorer leurs performances à partir de l'expérience (entraînement) sans être explicitement programmés. À plus grande échelle, l'apprentissage automatique est le processus qui consiste à enseigner aux systèmes informatiques comment faire des prédictions précises lors de la réception de données grâce à une analyse statistique et sans intervention ou assistance humaine.

L'objectif de l'apprentissage automatique est généralement de comprendre la structure des données et d'adapter ces données à des modèles qui peuvent être compris et utilisés par des personnes, en d'autres termes il permet d'apprendre ce qui se fait naturellement chez les humains et se révèle utile lorsque nous avons des tâches ou des problèmes complexes impliquant une grande quantité de données [7].

L'apprentissage automatique couvre de multiples applications, telles que la reconnaissance d'images, la reconnaissance vocale, le traitement du langage naturel, etc.

Tout comme l'écriture de code ordinaire, nous écrivons un algorithme, la machine exécute l'algorithme sur des données spécifiques, puis elle peut effectuer la même tâche avec de nouvelles données qu'elle n'a jamais vues auparavant. Cependant, au lieu d'écrire manuellement du code à l'aide d'un ensemble d'instructions spécifique, grâce à l'apprentissage automatique, les machines sont entraînées à l'aide de grandes quantités de données et apprennent à effectuer des tâches sans leur dire explicitement comment le faire.

Les ordinateurs doivent être formés pour atteindre leurs objectifs et, au cours du processus d'apprentissage, ils essaieront d'accéder à plus de données sur une période de temps pour créer une logique et l'améliorer.

L'apprentissage automatique peut se faire de plusieurs manières : il peut s'agir d'un apprentissage supervisé, d'un apprentissage non supervisé, d'un apprentissage semi-supervisé ou d'un apprentissage par renforcement.

2.3 Types de Machine learning

2.3.1 Apprentissage supervisé

L'apprentissage supervisé est l'un des types de machine learning les plus couramment utilisés. Dans ce cadre, l'algorithme apprend à partir d'un ensemble de données étiquetées, c'est-à-dire des données pour lesquelles les résultats souhaités sont déjà connus. L'objectif est de créer un modèle capable de prédire des résultats pour de nouvelles données [8].

Exemples d'algorithmes :

- Régression linéaire et logistique : Utilisées pour les prédictions continues et les classifications binaires.
- Arbres de décision : Utilisés pour la classification et la régression, offrant une visualisation intuitive des décisions.
- Forêts aléatoires : Combinaison de plusieurs arbres de décision pour améliorer la précision et réduire le surapprentissage.

- **Machines à vecteurs de support (SVM)** : Utilisées pour les problèmes de classification en maximisant la marge entre les classes.

2.3.2 Apprentissage non supervisé

L'apprentissage non supervisé traite des données qui ne sont pas étiquetées. L'objectif est de découvrir des structures cachées ou des motifs dans les données sans connaissance préalable des résultats.

Exemples d'algorithmes :

- **K-means** : Utilisé pour le clustering, où les données sont regroupées en clusters basés sur leurs similarités.
- **Analyse en composantes principales (PCA)** : Réduction de la dimensionnalité des données tout en conservant le plus d'information possible.
- **Algorithmes de clustering hiérarchique** : Créent une hiérarchie de clusters en fusionnant ou en divisant successivement les clusters.

2.3.3 Apprentissage par renforcement

L'apprentissage par renforcement (RL) est une méthode où un agent apprend à prendre des décisions en interagissant avec un environnement. L'agent reçoit des récompenses ou des punitions en fonction de ses actions et ajuste sa stratégie pour maximiser la récompense cumulée.

Exemples d'algorithmes :

Q-learning : Un algorithme basé sur des tables de valeurs pour apprendre la meilleure action à chaque état.

Deep Q-Network (DQN) : Utilise des réseaux de neurones pour approximer les valeurs des actions dans des environnements complexes.

2.3.4 Apprentissage semi-supervisé

L'apprentissage semi-supervisé combine des éléments des apprentissages supervisé et non supervisé. Il utilise une petite quantité de données étiquetées et une grande quantité de données non étiquetées pour former les modèles. Cela est particulièrement utile lorsque l'étiquetage des données est coûteux ou laborieux.

Exemples d'algorithmes :

- **Propagation de labels** : Utilise des graphes pour propager les labels connus aux points de données non étiquetés.
- **Auto-encodeurs semi-supervisés** : Réseaux de neurones qui apprennent à représenter les données de manière non supervisée tout en utilisant les étiquettes disponibles pour guider l'apprentissage.

2.4 Algorithmes de Machine Learning

Dans cette section, nous aborderons en détail les algorithmes de classification (apprentissage supervisé) et leurs formulations mathématiques.

2.4.1. Decision Tree (arbre de décision)

Un arbre de décision est un diagramme ou un graphique qui aide à déterminer un plan d'action ou à montrer une probabilité statistique. Le diagramme est appelé arbre de décision en raison de sa ressemblance avec la plante en question. Il se présente généralement sous la forme d'un diagramme vertical ou horizontal qui se ramifie. À partir de la décision elle-même (appelée "nœud"), chaque "branche" de l'arbre de décision représente une décision, un résultat ou une réaction possible. Les branches les plus éloignées de l'arbre représentent les résultats finaux d'une certaine voie de décision et sont appelées les "feuilles" [9].

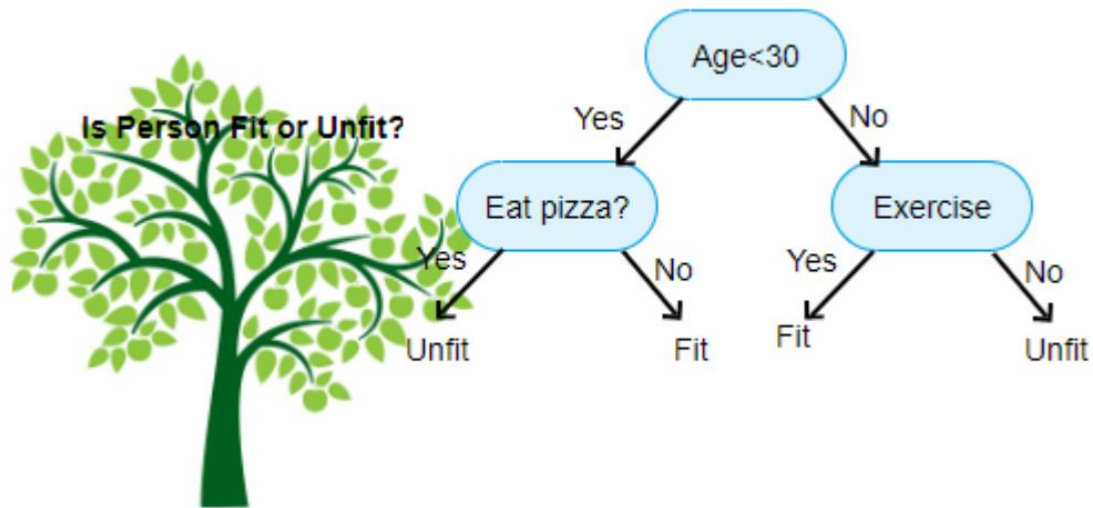


Figure 1 : Arbre de décision pour classification fit/unfit

L'arbre de décision est la mise en œuvre de la stratégie "diviser pour mieux régner" sur un ensemble d'instances indépendantes pour apprendre le problème. L'arbre de décision décisionnel est composé d'une racine, de nœuds de décision internes et de feuilles terminales. Chaque nœud d'un nœud de décision représente un test d'un attribut particulier ou une fonction d'un ou plusieurs attributs dans l'ensemble d'instances à classer. Le résultat du test représente des branches, ainsi chaque branche représente donc la valeur de test que le nœud peut prendre. Ce processus commence à la racine et est répété de manière récursive jusqu'à ce qu'un nœud feuille soit atteint, puis l'instance est classée selon la classe attribuée à la feuille.

En théorie des graphes, un arbre est un graphe non orienté, acyclique et connexe. L'ensemble des nœuds se divise en 3 catégories :

- Nœud racine : l'accès à l'arbre s'effectue par ce nœud •
- Nœud interne : les nœuds qui ont des descendants qui sont à leur tour des nœuds
- Nœuds terminaux (feuilles) : nœuds qui n'ont pas de descendant
- Branche : définit le résultat d'un test effectué sur les nœuds internes

Les gens utilisent les arbres de décision pour clarifier, cartographier et trouver une réponse à un problème complexe. Les arbres de décision sont fréquemment utilisés pour déterminer un plan d'action dans les domaines de la finance, de l'investissement ou des affaires. En mathématiques, les arbres de décision sont également appelés diagrammes en arbre.

Le problème à élucider avec les arbres de décision est de déterminer la répartition d'une population d'individus en groupes homogènes en fonction d'un ensemble de variables discriminantes et conformément à un objectif fixe qui est la variable cible.

Comme tout algorithme, les arbres de décision ont leurs points forts et faiblesses.

Les points forts des arbres de décision sont les suivants :

- Les arbres de décision sont capables de générer des règles compréhensibles.
- Les arbres de décision effectuent la classification sans nécessiter beaucoup de calculs.
- Les arbres de décision sont capables de traiter des variables continues et catégorielles.
- Les arbres de décision fournissent une indication claire des champs les plus importants pour la prédiction ou la classification.

Les faiblesses de la méthode sont :

- Les arbres de décision sont moins appropriés pour les tâches d'estimation où l'objectif est de prédire la valeur d'un attribut continu.
- Les arbres de décision sont sujets à des erreurs dans les problèmes de classification avec de nombreuses classes et un nombre relativement faible d'exemples d'apprentissage.
- La formation d'un arbre de décision peut être coûteuse en termes de calcul. Le processus de croissance d'un arbre de décision est coûteux en termes de calcul. À chaque nœud, chaque champ candidat à la division doit être trié avant que la meilleure division puisse être trouvée. Dans certains algorithmes, des combinaisons de champs sont utilisées et une recherche doit être effectuée pour trouver les poids de combinaison optimaux.

- Ils sont instables, c'est-à-dire que de petits changements dans les données peuvent produire des arbres très différents. Les modifications apportées aux nœuds proches de la racine peuvent grandement affecter l'arborescence résultante. Nous disons alors que les arbres produisent des estimateurs de variance élevée.

2.4.2. Random Forest

Le Random Forest est un Algorithme de classification composé de nombreux arbres de décisions. Formellement proposé en 2001 par Leo Breiman et Adèle Cutler, il fait partie des techniques d'apprentissage automatique. Cet algorithme combine les concepts de sous-espaces aléatoires et de rééchantillonnage avec remise ensembliste (bagging). L'algorithme des forêts d'arbres décisionnels effectue un apprentissage sur de multiples arbres de décision entraînés sur des sous-ensembles de données légèrement différents.

La forêt aléatoire consiste en un ensemble d'arbres de décision binaires qui introduisent le caractère aléatoire. Ces arbres se dressent distinguer les uns des autres par les sous-échantillons de données sur lesquels ils sont entraînés. Ces sous-échantillons sont tirés au hasard (d'où le terme « aléatoire ») dans un jeu de données.

La technique des forêts aléatoires modifie la méthode du Bagging appliquée aux arbres en ajoutant un critère de décorrélation entre ces arbres. L'idée de cette méthode est de réduire la corrélation sans augmenter trop la variance. Le principe consiste à choisir de façon aléatoire un sous-ensemble de variables qui sera considéré à chaque niveau de choix du meilleur nœud de l'arbre [10].

Principe de l'algorithme

Considérons un ensemble d'entraînement $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$, et a le nombre d'attributs des exemples de X (individus).

Considérons \mathbf{St} un bootstrap contenant m instances obtenues par rééchantillonnage avec remplacement de \mathbf{S} et soit $\{\mathbf{h1}, \dots, \mathbf{ht}\}$ un ensemble de \mathbf{T} arbres de décision tels que chaque arbre \mathbf{ht} est construit à partir de \mathbf{St} .

Pour chaque nœud de l'arbre, l'attribut de partitionnement est choisi en considérant un nombre f ($f < a$) d'attributs choisis aléatoirement (parmi les attributs). Celui choisi c'est celui qui va optimiser le critère d'homogénéité considéré par les arbres utilisés (entropie de Shannon ou indice de Gini)

Pour classifier une nouvelle instance, le classificateur des forêts aléatoires effectue un vote de majorité uniformément pondéré des classificateurs de cet ensemble.

Comme tout algorithme, les Random Forest ont leurs points forts et faiblesses.

Les points forts des arbres de décision sont les suivants :

- Elles permettent d'éviter le sur-apprentissage.
- Elles permettent d'améliorer les performances des arbres de décision.

Les faiblesses de la méthode sont :

- La perte de lisibilité des arbres de décisions.
- Importants temps de calcul.

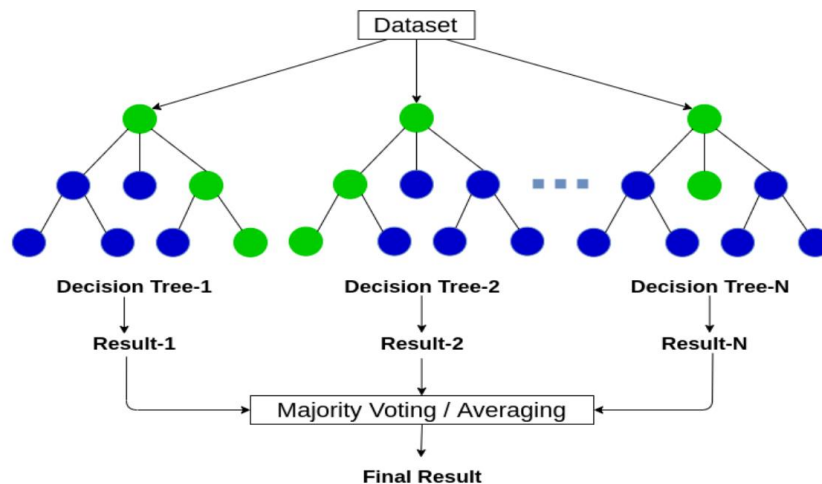


Figure 2 : Exemple illustrant le déroulement de Random Forest

2.4.3. Régression linéaire et logistique (Logistic regression)

La régression logistique ou modèle logit est une méthode statistique puissante qui peut modéliser des résultats binomiaux avec une ou plusieurs variables explicatives. Il mesure la relation entre une variable dépendante catégorielle et une ou plusieurs variables indépendantes en utilisant une fonction logistique (c'est-à-dire une distribution logistique cumulative) pour estimer la probabilité.

La régression logistique a été utilisée dans les sciences biologiques au début du vingtième siècle. Elle a ensuite été utilisée dans de nombreuses applications des sciences sociales. La régression logistique est utilisée lorsque la variable dépendante (cible) est catégorique.

Par exemple : Pour prédire si un email est un spam (1) ou (0)

Si la tumeur est maligne (1) ou non (0)

La régression logistique indique généralement où se trouve la frontière entre les différentes classes, en plus de ça elle indique que les probabilités des classes dépendent de la distance de la frontière.

La régression logistique est une méthode prédictive. Cependant, par régression logistique, cette prédiction conduira à une dichotomie. La régression logistique est l'un des outils les plus couramment utilisés en statistiques appliquées et analyse de données discrètes [11].

La fonction qui régit le modèle de régression logistique est le suivant :

$$P = \frac{1}{1 + e^{-ywx}}$$

Tel que : x est le vecteur de la donnée où $x_i \in \mathbb{R}^n$, y est le vecteur de l'étiquette de la classe où $y_i \in \{1,-1\}$ et $w \in \mathbb{R}^n$ est le vecteur des poids.

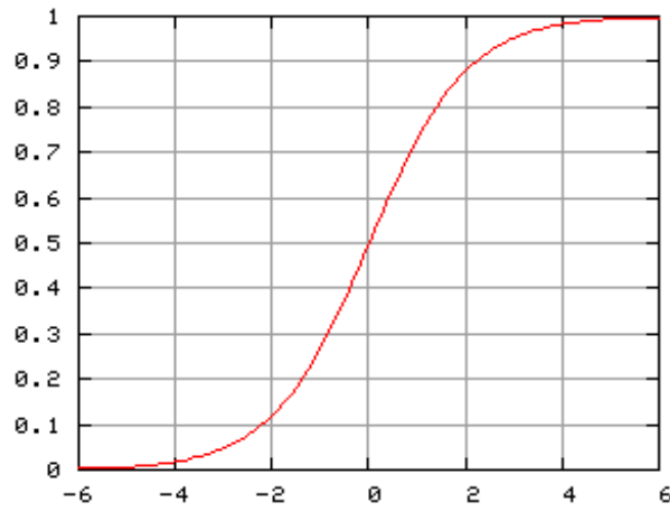


Figure 3 : Variations de la fonction logistique

2.4.4. Support Vector Machine (SVM)

La machine à vecteurs de support (SVM) a été introduite par Vapnik (1995). Il s'agit d'un mariage entre la modélisation linéaire et l'apprentissage basé sur les instances. Il sélectionne un petit nombre d'instances limites critiques, appelées vecteurs de support, dans chaque classe et construit une fonction discriminante linéaire qui sépare chaque classe aussi largement que possible. Le système transcende les limites des frontières linéaires en rendant pratique l'inclusion de termes non linéaires dans la fonction, ce qui permet de former des frontières de décision quadratiques, cubiques et d'ordre supérieur.

Soit un ensemble de points de 2 types dans N lieu dimensionnel, SVM génère un hyperplan dimensionnel (N - 1) pour séparer ces points en 2 groupes. Supposons que certains points de 2 types peuvent être séparés linéairement. SVM trouvera la ligne droite qui sépare ces points en 2 types et qui se situe le plus loin possible de tous ces points et ce problème est dit linéairement séparable sinon il n'est pas linéairement séparable et il n'existe pas un hyperplan séparable.

L'idée de base du SVM est d'utiliser un modèle linéaire pour mettre en œuvre des limites de classe non linéaires par le biais d'une mise en correspondance non linéaire du vecteur d'entrée

dans un espace de caractéristiques de haute dimension. Un modèle linéaire construit dans le nouvel espace peut représenter une limite de décision non linéaire dans l'espace original. Dans le nouvel espace, un hyperplan de séparation optimal est construit. Ainsi, le SVM est connu comme l'algorithme qui trouve un type spécial de modèle linéaire, l'hyperplan de la marge maximale. L'hyperplan de marge maximale donne la séparation maximale entre les classes de décision. Les exemples d'apprentissage qui sont les plus proches de l'hyperplan de la marge maximale sont appelés vecteurs de support. Tous les autres exemples de formation ne sont pas pertinents pour définir les limites des classes binaires.

Le SVM repose donc sur deux notions principales qui sont : la notion de marge maximale et la notion de fonction noyau.

Les machines à vecteurs de support, comme les réseaux neuronaux, ne subissent pas les contraintes des distributions statistiques. Avec les machines à vecteurs de support, il est peu probable qu'il y ait surajustement et elles produisent souvent des classificateurs très précis. En revanche, leur calcul est très complexe et elles sont lentes par rapport à d'autres algorithmes d'apprentissage automatique lorsqu'elles sont appliquées dans un cadre non linéaire [12].

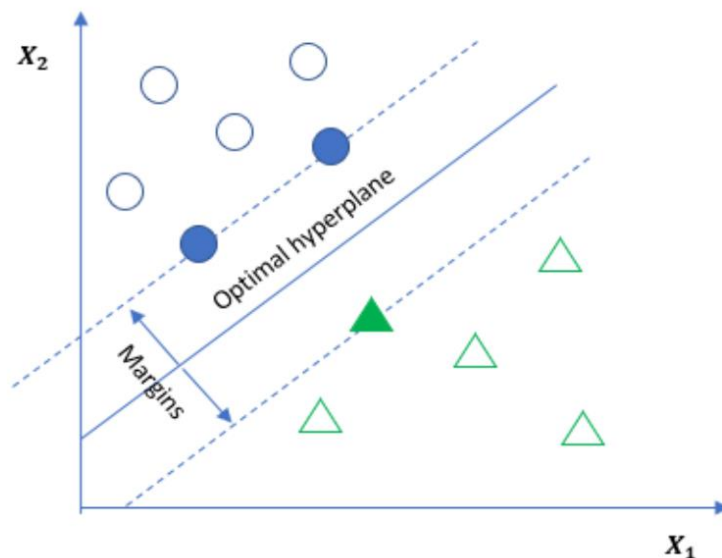


Figure 4 : Fonctionnement du modèle Support Vector Machine

Comme tout algorithme, les SVM ont leurs points forts et faiblesses.

Ses points forts sont les suivants :

- Elles sont très efficaces du fait qu'elles utilisent un sous-ensemble de points d'entraînement.
- Elles reposent sur une solide base théorique

Les faiblesses de la méthode sont :

- La complexité des formules mathématiques utilisées lors de la classification.
- Un temps d'entraînement très long, ce qui fait qu'elles ne conviennent pas à des jeux de données volumineux.

2.4.5. KNN (k-Nearest-Neighbors)

L'algorithme K-Nearest Neighbors (KNN) est l'une des méthodes de classification les plus fondamentales et les plus simples, basée sur les exemples d'apprentissage les plus proches dans l'espace des caractéristiques. K-NN est un type d'algorithme basé sur l'instance dans la catégorie des algorithmes d'apprentissage paresseux (Aha, 1997). K-NN classe un objet en fonction de sa similarité avec d'autres objets. La logique suppose que les objets similaires sont proches les uns des autres et que les objets dissemblables sont éloignés les uns des autres. Un objet est donc étiqueté en fonction de l'étiquette de la majorité de ses voisins. La similarité des objets est évaluée à l'aide d'une métrique de distance appropriée, généralement la distance euclidienne. La distance euclidienne est utilisée comme métrique de distance pour les variables continues. Cependant, il n'existe pas de concept commun pour définir le nombre de voisins les plus proches, il est défini afin d'obtenir une bonne précision de classification, mais il est intuitif d'utiliser plus d'un voisin le plus proche si la taille de l'ensemble d'apprentissage est grande.

Cette méthode simple présente quelques problèmes pratiques : elle a tendance à être lente pour les grands ensembles d'apprentissage, elle est peu performante avec les données bruyantes et elle est peu performante avec les attributs non pertinents car chaque attribut a la même influence sur la décision, tout comme dans la méthode. D'autre part, l'avantage de cette méthode

simple par rapport à la plupart des autres méthodes d'apprentissage automatique est qu'elle permet d'ajouter de nouveaux exemples à l'ensemble d'apprentissage à tout moment [13].

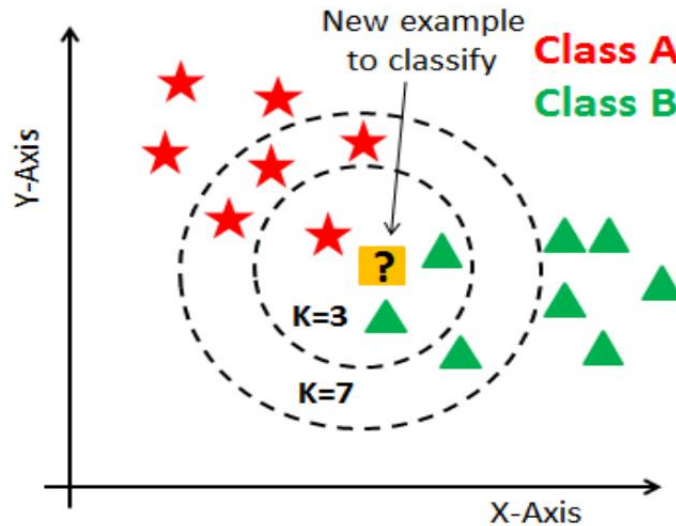


Figure 5 : Exemple de classification KNN ($K=3$ et $K=5$)

2.4.6 Réseaux de neurones artificiels

Le réseau neuronal artificiel (ANN) est la pièce d'un système informatique conçu pour simuler la façon dont le cerveau humain analyse et traite l'information. L'unité de base du calcul dans un réseau neuronal est le neurone, souvent appelé nœud ou unité. Il reçoit des données en provenance d'autres nœuds ou d'une source externe et calcule une sortie. La figure 1[a] montre un neurone biologique et un neurone artificiel. Un type courant et simple de réseau neuronal est le réseau neuronal à action directe. Il contient plusieurs neurones, disposés en couches. Les nœuds des couches adjacentes sont reliés entre eux par des connexions ou des arêtes. Toutes ces connexions sont associées à des poids. Un exemple est illustré à la figure 1[b].

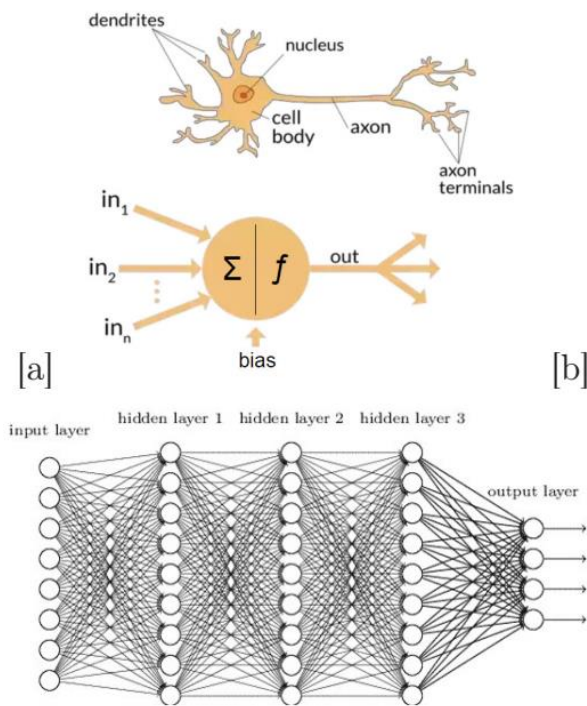


Figure 6 : Visualisation (a) des neurones biologiques et artificiels et (b) un exemple d'un réseau neuronal

Il existe une multitude d'architectures de réseaux de neurones, bien que la structure fondamentale de ces modèles soit la même, chaque architecture présente des mécanismes de raisonnement internes différents qui méritent d'être explorés et sont souvent utilisés dans une application spécifique. Les modèles de réseaux de neurones les plus connus sont les suivants : Le perceptron multicouche (MLP) qui est le modèle le plus basique et qui n'a pas connu de succès pendant longtemps, car il reste généralement bloqué dans un optimum local lors de la formation, le réseau neuronal convolutif (CNN) est surtout utilisé pour la reconnaissance faciale- la recherche et l'édition d'images, le réseau neuronal récurrent (RNN) [14] est appliqué à la traduction automatique, à la reconnaissance vocale et à la prédiction de séries temporelles. L'autocodeur (AE) [15] est un célèbre modèle de réduction de la dimensionnalité et les réseaux adversariaux génératifs (GAN) [16] sont utilisés pour générer de nouvelles instances synthétiques de données qui peuvent passer pour des données réelles [17].

2.5. Evaluation et validation

Vous devez toujours évaluer le modèle pour déterminer s'il vous aidera à prédire correctement la cible dans les nouvelles données à venir. Étant donné que la future instance a une valeur cible inconnue, vous devez vérifier l'indice de précision du modèle ML sur les données qui connaissent déjà la réponse cible, puis utiliser ce niveau comme indice de la précision prédictive des données futures.

La mesure de précision choisie influe sur l'applicabilité des modèles et sur leurs performances hors échantillon.

2.6. Mesure de performance

2.6.1. Confusion Matrix (matrice de confusion)

En classification, la matrice d'évaluation habituelle est la matrice de confusion. C'est un tableau qui va décrire les performances d'un modèle et mesurer sa qualité sur un ensemble de données d'entraînement, Dans la matrice, les prédictions absolues sont divisées en prédictions correctes et fausses, c'est-à-dire le nombre de prédictions correctes et fausses établies par le modèle de classification par rapport aux résultats réels (valeur cible) dans les données. La figure suivante affiche une matrice de confusion pour deux classes (2x2).

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 7 : Confusion Matrix

- TP (True Positive) : Vrai positifs, c'est-à-dire les cas où la prédiction est positive, et la valeur réelle est aussi positive.
- TN (True Negative) : Vrai négatif, c'est-à-dire les cas où la prédiction est négative, et la valeur réelle est aussi négative.
- FP (False Positive) : Faux positif, c'est-à-dire les cas où la prédiction est positive, mais la valeur réelle est négative.
- FN (False Negative) : Faux négatif, c'est-à-dire les cas où la prédiction est négative, mais la valeur réelle est positive.

2.6.2. Métriques d'évaluation

Maintenant que nous avons les 4 valeurs TP, TN, FP, FN qui sont nécessaire mais pas suffisante pour juger un modèle, c'est pour ça qu'on devrait calculer d'autres métriques qui nous permettront de mesurer la performance du modèle comme il se doit.

TPR (True Positive Rate) - La Sensibilité

C'est le taux des vrais positifs, c'est-à-dire la proportion de cas positifs correctement identifiés, calculée par la formule suivante :

$$TPR = \frac{TP}{TP + FN}$$

TNR (True Negative Rate) - La Spécificité

C'est le taux des vrais négatifs, c'est-à-dire la proportion de cas négatifs correctement identifiés, calculée par la formule suivante :

$$TNR = \frac{TN}{TN + FP}$$

FPR (False Positif Rate) – Erreur du Type I

C'est le taux des faux positifs, c'est-à-dire la proportion de cas négatifs non identifiés ou en d'autres termes c'est les cas négatifs qui ont été classé incorrectement comme cas positifs, Calculée par la formule suivante :

$$FPR = \frac{FP}{TN + FP}$$

FNR (False Negative Rate) – Erreur du Type II

C'est le taux des faux négatifs, c'est-à-dire la proportion de cas positifs non identifiés ou en d'autres termes c'est les cas positifs qui ont été classé incorrectement comme cas négatifs, calculée par la formule suivante :

$$FNR = \frac{FN}{TP + FN}$$

La précision

La précision représente sur tout l'ensemble des points qui sont déclarés positifs, le pourcentage de ces points qui sont réellement positifs, calculée par la formule suivante :

$$\text{Précision} = \frac{TP}{TP + FP}$$

F1-Score (Le F-mesure)

Il est utilisé pour mesurer la qualité du test. Il s'agit d'une moyenne pondérée de la précision et de la sensibilité. Lorsque le score F1 est de 1, c'est le meilleur score, tandis que lorsqu'il est égal à 0, c'est le pire.

La précision et la sensibilité se font généralement au détriment l'une de l'autre. Autrement dit, une grande précision se fait au détriment de la sensibilité, et une grande sensibilité au détriment de la précision. Le modèle idéal doit présenter une sensibilité et une précision élevées, et c'est là que le F1-Score intervient pour trouver l'harmonie idéal entre ces deux métriques pour avoir un modèle robuste.

Le F1-Score est calculé par la formule suivante :

$$\text{F1 - Score} = \frac{2 \times (\text{précision} \times \text{sensibilité})}{(\text{précision} + \text{sensibilité})}$$

2.6.3. Courbes d'évaluation-Receiver Operating Characteristics Curve (ROC Curve)

« La fonction d'efficacité du récepteur, plus fréquemment désignée sous le terme -courbe ROC-dite aussi caractéristique de performance (d'un test) ou courbe sensibilité/spécificité, est une mesure de la performance d'un classificateur binaire, c'est-à-dire d'un système qui a pour objectif de catégoriser des éléments en deux groupes distincts sur la base d'une ou plusieurs des caractéristiques de chacun de ces éléments. Graphiquement, on représente souvent la mesure ROC sous la forme d'une courbe qui donne le taux de vrais positifs (TPR) en fonction du taux de faux positifs (FPR).

Les courbes ROC furent inventées pendant la Seconde Guerre mondiale pour montrer la séparation entre les signaux radar et le bruit de fond. » [18]

La courbe ROC nous permettra de visualiser le compromis entre spécificité et sensibilité en représentant graphiquement l'évolution de la (sensibilité) en fonction de (1-spécificité) selon les valeurs d'un certain seuil S (Threshold).

Diminuer la valeur du seuil de classification S va permettre de classer plus d'éléments comme positifs, ce qui va augmenter le nombre de faux positifs et de vrais positifs.

Soit x un individu et soient les fonctions suivantes :

- La sensibilité $\alpha(S) = \text{prob}(\text{score}(x) \geq S \mid x = \text{événement})$, c'est-à-dire bien détecter un événement au seuil S .
- La spécificité = $\text{prob}(\text{score}(x) \leq S \mid x = \text{non-événement})$, c'est-à-dire qui implique de bien détecter un non-événement au seuil S .

On dira alors que la proportion des non- événement déclarés comme événement est $1 - \beta(S)$.

La courbe ROC va donc représenter $\alpha(S)$ en fonction de $1 - \beta(S)$ pour des valeurs de S allant du :

- Maximum où l'on considère tous les individus comme non-événement ce qui implique que :

$$\alpha(S) - 1 - \beta(S) = 0.$$
- Minimum où l'on considère tous les individus comme événement ce qui implique que :

$$\alpha(S) - 1 - \beta(S) = 1$$

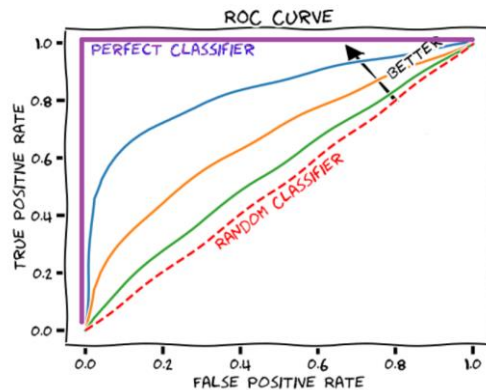


Figure 8 : Courbe ROC

Interprétation des points critiques dans le graphe précédent

- Au point (0, 0) le classificateur déclare tous les individus comme non-événement : c'est à dire il n'y a aucun faux positif, mais également aucun vrai positif.
- Au point (1, 1) le classificateur déclare tous les individus comme événement : c'est-à dire il n'y a aucun vrai négatif, mais également aucun faux négatif.

- Au point (0, 1) le classificateur n'a aucun faux positif ni aucun faux négatif, et est par conséquent parfaitement exact, c'est-à-dire ne se trompant jamais.
- Au point (1, 0) le classificateur n'a aucun vrai négatif ni aucun vrai positif, et est par conséquent parfaitement inexact, c'est-à-dire se trompant toujours.

2.6.4. Aire sous la courbe ROC (Area Under Curve - AUC)

AUC - La courbe ROC est une mesure de performance pour les problèmes de classification à différents seuils. La courbe ROC étant une courbe de probabilité, l'AUC représente le degré ou la mesure de la séparabilité. Elle indique dans quelle mesure le modèle est capable de faire la distinction entre les classes. Plus l'AUC est élevée, plus le modèle est capable de prédire les 0 comme 0 et les 1 comme 1. Par exemple, plus l'AUC est élevée, plus le modèle est capable de distinguer les patients atteints de la maladie de ceux qui ne le sont pas.

Plus précisément, cette aire est la probabilité que le score d'un individu x tiré aléatoirement de l'ensemble des individus libellés comme événement soit supérieur au score d'un individu y tiré aléatoirement de l'ensemble des individus libellés comme non-événement. Si l'air est égal à 1, cela veut dire que tous les scores des individus x sont supérieurs aux scores des individus y [19].

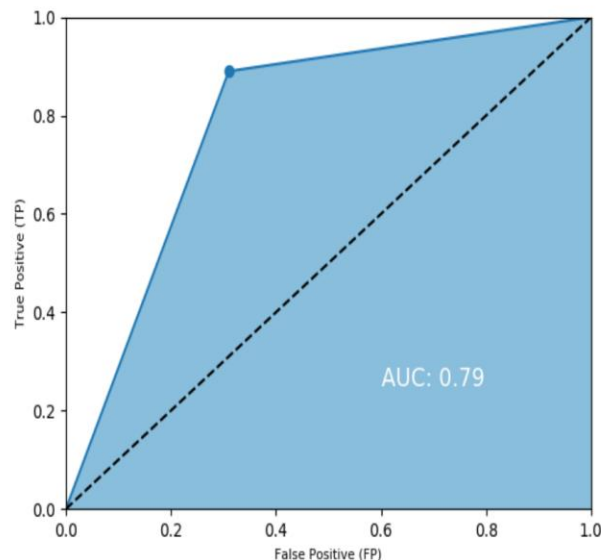


Figure 9 : AUC

Lors de la création et de l'entraînement d'un modèle de machine learning, l'objectif est de choisir le modèle qui fait les meilleures prédictions, c'est-à-dire de choisir le modèle avec les meilleurs paramètres (paramètres ou hyperparamètres du modèle de machine learning).

Cependant, si vous choisissez des paramètres de modèle qui produisent les « meilleures » performances prédictives sur les données de l'évaluation, vous risquez de sur ajuster le modèle. Lorsque le modèle mémorise les tendances qui apparaissent dans les sources de données de formation et d'évaluation, mais ne parvient pas à généraliser ces tendances dans les données, un surapprentissage se produit. Cela se produit généralement lorsque les données d'entraînement incluent toutes les données utilisées dans l'évaluation. Les modèles de surapprentissage ont bien fonctionné pendant la période d'évaluation, mais n'ont pas été en mesure de faire des prédictions précises sur des données inconnues.

Pour éviter de sélectionner un modèle sur ajusté comme meilleur modèle, vous pouvez conserver d'autres données pour vérifier les performances du modèle ML. Par exemple, vous pouvez séparer les données en utilisant 60 % pour la formation, 20 % pour l'évaluation et 20 % pour la validation.

Cependant, l'utilisation des données du processus de formation pour l'évaluation et la vérification réduira la quantité de données disponibles pour la formation. Ceci est particulièrement problématique pour les petits ensembles de données, car il est préférable d'utiliser autant de données que possible pour la formation. Pour résoudre ce problème, vous pouvez effectuer une validation croisée.

2.6.5. Cross Validation (CV)

La validation croisée (CV) est importante pour garantir la bonne validité des modèles créés. Cependant, elle n'a pas toujours été aussi importante ou réalisable. Les premières adoptions consistaient généralement à tester et à entraîner le modèle sur les mêmes données, car les informations et la puissance de calcul étaient rares. Il s'agit là d'un cas classique de surajustement du modèle ou, en d'autres termes, l'adaptation du modèle au point que les nouvelles observations introduites seraient très probablement faussement classées. La première tentative

de validation croisée primitive a été introduite avec la division stricte entre les données de test et de formation. Cela réduit le problème de l'overfitting et crée donc des modèles plus robustes qui conservent un pouvoir prédictif en dehors de l'environnement d'apprentissage. Cette méthode est encore utilisée aujourd'hui, mais des méthodes plus avancées de CV sont souvent préférées.

La validation croisée à k blocs, « k-fold cross-validation » : elle consiste à diviser l'échantillon original en k échantillons (ou « blocs »), puis sélectionner un des k échantillons comme ensemble de validation pendant que les k-1 autres échantillons constituent l'ensemble d'apprentissage. Après apprentissage, on peut calculer la performance de validation déjà vu auparavant. Puis on répète l'opération en sélectionnant un autre échantillon de validation parmi les blocs prédéfinis. À l'issue de la procédure nous obtenons ainsi k scores de performances, un par bloc. La moyenne et l'écart type des k scores de performances peuvent être calculés pour estimer le biais et la variance (expliqués en annexe) de la performance de validation. [20]

Le tableau suivant illustre le fonctionnement d'une validation croisée à 3 blocs :

k	Bloc 1	Bloc 2	Bloc 3
1	Validation	Apprentissage	Apprentissage
2	Apprentissage	Validation	Apprentissage
3	Apprentissage	Apprentissage	Validation

Tableau 1: Le fonctionnement d'une validation croisée à 3 blocs

Chapitre 3: Machine learning dans le domaine des applications mobiles pour la gestion budgétaire

3.1 Introduction

Dans ce chapitre, nous introduisons le domaine de l'apprentissage automatique (ML) et son application dans le domaine de la gestion budgétaire. Nous explorons comment les techniques d'apprentissage automatique peuvent révolutionner les prédictions financières, les recommandations et les plans budgétaires. Le chapitre offre un aperçu des concepts clés, des techniques et des défis liés à l'application de l'apprentissage automatique à la gestion budgétaire.

3.2 Qu'est-ce que l'apprentissage automatique et comment s'applique-t-il à la gestion budgétaire ?

L'apprentissage automatique est une branche de l'intelligence artificielle qui se concentre sur le développement d'algorithmes et de modèles capables d'apprendre à partir de données et de prendre des décisions ou de faire des prédictions sans être explicitement programmés [3]. Dans le contexte de la gestion budgétaire, l'apprentissage automatique joue un rôle crucial en utilisant des approches basées sur les données pour optimiser les budgets, personnaliser les plans financiers et améliorer la santé financière globale.

Les algorithmes d'apprentissage automatique apprennent des modèles et des relations à partir de données financières, telles que les enregistrements de dépenses, les mesures biométriques et les préférences des utilisateurs. En analysant ces données, les modèles d'apprentissage automatique peuvent extraire des insights significatifs et faire des prédictions qui aident les individus à atteindre leurs objectifs financiers.

Par exemple, les algorithmes d'apprentissage automatique peuvent prédire les dépenses mensuelles en fonction de caractéristiques telles que les types de dépenses, les habitudes d'achat, et les tendances historiques. Ces informations aident les individus à suivre précisément leurs dépenses, à prendre des décisions éclairées et à planifier efficacement leur budget. De plus, l'apprentissage automatique peut être utilisé pour classifier les différents types de dépenses, détecter les anomalies dans les habitudes de dépenses, recommander des combinaisons budgétaires optimales, ou estimer les niveaux de dépenses mensuelles. En intégrant l'apprentissage automatique dans les applications de gestion budgétaire, les individus peuvent

bénéficier de recommandations personnalisées, d'analyses basées sur les données, et de plans financiers optimisés adaptés à leurs besoins spécifiques et à leurs objectifs financiers.

Les modèles d'apprentissage automatique peuvent s'adapter et s'améliorer au fil du temps à mesure que plus de données deviennent disponibles, permettant un affinement continu des stratégies financières et aidant les individus à mener une vie financière plus saine et plus équilibrée.

3.3 Comment l'apprentissage supervisé peut-il améliorer les prédictions financières ?

L'apprentissage supervisé, une technique d'apprentissage automatique, consiste à entraîner des algorithmes à l'aide de données annotées pour faire des prédictions ou des classifications. Dans le contexte de la gestion budgétaire, l'apprentissage supervisé peut considérablement améliorer les prédictions financières en utilisant des données historiques et des résultats connus.

Les modèles d'apprentissage supervisé peuvent être entraînés à l'aide de données financières annotées, où les données d'entrée comprennent des caractéristiques telles que les types de dépenses, les revenus mensuels, les dettes accumulées et les labels de sortie correspondants représentent les prévisions souhaitées ou les classifications. Ces modèles apprennent à partir de ces données annotées pour faire des prévisions précises ou des classifications pour de nouvelles données non vues.

Les algorithmes d'apprentissage supervisé dans le domaine de la gestion budgétaire peuvent offrir divers avantages. Par exemple, ils peuvent prédire les dépenses mensuelles en fonction des caractéristiques d'entrée telles que les revenus mensuels, les types de dépenses et les habitudes d'achat [6]. En estimant précisément les dépenses mensuelles, les individus peuvent mieux suivre leurs finances, planifier leur budget et optimiser leurs dépenses pour atteindre leurs objectifs financiers.

Une autre application est la classification des différents types de dépenses, où les algorithmes d'apprentissage supervisé peuvent classifier les différentes catégories de dépenses en fonction de caractéristiques d'entrée telles que les tendances historiques de dépenses et les habitudes

d'achat. Cette classification peut aider les individus à suivre la diversité de leurs dépenses, à diversifier leurs budgets et à s'assurer d'une gestion équilibrée de leurs finances.

L'apprentissage supervisé peut également être utilisé pour estimer les niveaux de revenus et de dettes, permettant aux individus de mesurer le niveau d'effort nécessaire pour gérer leurs finances. En analysant les caractéristiques d'entrée telles que les revenus mensuels, les niveaux d'endettement et les habitudes de consommation, les modèles d'apprentissage supervisé peuvent prédire les niveaux de revenus et de dettes, aidant les individus à optimiser leur budget pour atteindre les résultats financiers souhaités.

Grâce à l'utilisation de l'apprentissage supervisé, les individus peuvent bénéficier de recommandations personnalisées, de prévisions précises et de prises de décision éclairées. Ces modèles permettent aux individus de suivre leurs progrès, d'optimiser leur budget et d'atteindre leurs objectifs financiers de manière plus efficace.

3.4 Quelles perspectives l'apprentissage non supervisé peut-il offrir dans le domaine de la gestion du budget ?

L'apprentissage non supervisé, une approche de l'apprentissage machine, offre des perspectives précieuses dans le domaine de la gestion du budget en analysant des données non étiquetées pour découvrir des motifs, des relations et des structures. Dans le contexte de la gestion du budget, les techniques d'apprentissage non supervisé peuvent fournir plusieurs perspectives qui aident à diversifier les approches de gestion, à identifier des regroupements d'éléments de dépenses, et à explorer de nouvelles stratégies d'optimisation financière.

Une application de l'apprentissage non supervisé dans la gestion du budget est le regroupement des éléments de dépenses. En analysant les données financières, les algorithmes d'apprentissage non supervisé peuvent identifier des groupes naturels ou des clusters d'éléments de dépenses en fonction de similarités dans les motifs de dépenses, les catégories d'achats, ou d'autres caractéristiques pertinentes. Cette information aide les individus à comprendre les relations entre les catégories de dépenses et à découvrir des variations qui ciblent des groupes spécifiques d'achats ou qui visent différents objectifs financiers. En diversifiant leurs stratégies de gestion

avec des approches provenant de différents clusters, les individus peuvent optimiser leur gestion budgétaire et éviter les inefficacités.

Une autre perspective offerte par l'apprentissage non supervisé est la détection d'anomalies dans les données financières. En apprenant les motifs présents dans un ensemble de données, les modèles d'apprentissage non supervisé peuvent identifier des schémas de dépenses inhabituels ou anormaux qui s'écartent de la norme [22]. Cette capacité est particulièrement utile pour identifier les erreurs potentielles dans la gestion des finances, détecter les irrégularités dans les dépenses, ou signaler des modèles de dépenses anormaux pouvant indiquer des risques financiers ou des opportunités d'économies. En détectant les anomalies, les individus peuvent ajuster leur gestion budgétaire et améliorer leur efficacité financière.

De plus, l'apprentissage non supervisé peut être utilisé pour la réduction de la dimensionnalité dans l'analyse des données financières. Les ensembles de données financières contiennent souvent de nombreuses variables, ce qui rend difficile leur interprétation et leur analyse. Des techniques d'apprentissage non supervisé telles que l'analyse en composantes principales (PCA) ou t-SNE (t-Distributed Stochastic Neighbor Embedding) peuvent réduire la dimensionnalité des données tout en conservant les informations importantes. Cela permet de visualiser les relations entre les catégories de dépenses, d'identifier des motifs et d'obtenir une compréhension plus approfondie des données [22].

Intégrer l'apprentissage non supervisé dans les applications de gestion du budget offre aux individus des perspectives précieuses pour améliorer leurs stratégies financières. En exploitant le regroupement des éléments de dépenses, la détection d'anomalies et la réduction de la dimensionnalité, les gestionnaires peuvent explorer de nouvelles approches, diversifier leurs stratégies de gestion budgétaire et obtenir une compréhension complète de leurs données financières.

3.5 Comment l'apprentissage par renforcement peut-il optimiser la gestion du budget ?

L'apprentissage par renforcement, une méthode d'apprentissage machine, offre des capacités uniques pour améliorer la gestion du budget en adaptant dynamiquement les stratégies financières en fonction des retours d'information en temps réel. Il permet à un agent, tel qu'une application de gestion budgétaire ou un outil financier, d'apprendre et de prendre des décisions pour maximiser les récompenses à long terme ou les résultats financiers.

Dans le domaine de la gestion du budget, l'apprentissage par renforcement peut être utilisé de plusieurs façons pour optimiser les plans financiers et améliorer les niveaux de gestion. Voici quelques aspects clés :

Adaptation Personnalisée : Les agents d'apprentissage par renforcement peuvent ajuster continuellement les stratégies budgétaires en fonction des progrès individuels, des préférences et des retours d'information en temps réel. En collectant des données sur les performances, telles que l'exécution des budgets, les variations de revenus ou les niveaux de dépenses, l'agent peut ajuster les paramètres budgétaires tels que la répartition des fonds, l'intensité des économies, la durée des investissements ou les périodes de révision pour maximiser les gains financiers pour chaque individu.

Optimisation Orientée vers les Objectifs : L'apprentissage par renforcement permet à l'agent d'optimiser la gestion budgétaire en fonction d'objectifs spécifiques. En définissant des objectifs financiers tels que l'épargne, l'investissement ou la réduction de la dette, l'agent peut apprendre et ajuster les plans pour atteindre ces objectifs. L'agent peut dynamiquement allouer les types de dépenses, les intensités d'investissement et les fréquences de répartition pour optimiser la progression vers les objectifs financiers désirés.

Planification Dynamique : Les agents d'apprentissage par renforcement peuvent utiliser des techniques telles que l'itération de valeur, l'apprentissage Q ou les gradients de politique pour prendre des décisions éclairées sur la séquence des budgets, le nombre d'allocations ou l'intensité des investissements. En tenant compte de la situation financière actuelle, des

performances passées et des résultats souhaités, l'agent peut planifier et ajuster dynamiquement les stratégies budgétaires pour optimiser l'expérience globale de gestion.

Feedback Adaptatif : Les agents d'apprentissage par renforcement peuvent fournir un feedback adaptatif et des conseils pendant la gestion budgétaire. En analysant les données en temps réel à partir de capteurs ou d'applications financières, l'agent peut offrir des conseils sur la répartition des fonds, les ajustements d'investissement ou les stratégies d'épargne pour améliorer l'expérience de gestion et garantir des performances optimales.

En utilisant les techniques d'apprentissage par renforcement, les individus peuvent bénéficier de plans financiers personnalisés et optimisés. Ces plans s'adaptent aux progrès individuels, tiennent compte des objectifs personnels et fournissent des conseils en temps réel. L'apprentissage par renforcement permet aux individus de maximiser leurs résultats financiers, d'améliorer leur gestion et de maintenir une gestion budgétaire saine et durable.

3.6 Quels sont les défis et limites de l'application de l'apprentissage machine dans la gestion du budget ?

Bien que l'apprentissage machine offre un potentiel significatif pour améliorer les applications liées à la gestion du budget, plusieurs défis et limitations doivent être pris en compte. Ces défis incluent la qualité des données, le surajustement, l'interprétabilité, les ressources computationnelles et les considérations éthiques. Comprendre et aborder ces défis est essentiel pour une application réussie de l'apprentissage machine dans le domaine de la gestion du budget.

Qualité des données : Obtenir des données de gestion du budget de haute qualité peut être difficile. Les données financières nécessitent souvent des capteurs ou des dispositifs spécialisés pour capturer avec précision les métriques pertinentes. Assurer l'exactitude, la fiabilité et la représentativité des données est essentiel pour former des modèles d'apprentissage machine fiables.

Surajustement : Le surajustement se produit lorsqu'un modèle d'apprentissage machine devient trop spécifique aux données d'entraînement et ne généralise pas bien aux données non vues. Dans le contexte de la gestion du budget, le surajustement peut conduire à des recommandations

ou prédictions sous-optimales qui ne s'appliquent pas à des individus ayant des caractéristiques financières différentes ou des contextes budgétaires variés. Équilibrer la complexité du modèle et sa généralisation est crucial pour surmonter ces défis de surajustement.

Interprétabilité : L'interprétabilité des modèles d'apprentissage machine est une préoccupation dans le domaine de la gestion du budget. Bien que les modèles complexes puissent atteindre une haute précision, leurs processus décisionnels peuvent être difficiles à interpréter. Des modèles interprétables permettent aux utilisateurs de comprendre pourquoi certaines prédictions ou recommandations sont faites, ce qui permet aux individus de faire confiance et d'agir sur les résultats du modèle.

Ressources computationnelles : Mettre en œuvre des modèles d'apprentissage machine dans des environnements contraints en ressources, tels que des applications de gestion de budget ou des dispositifs portables, nécessite une attention particulière aux ressources computationnelles. Les modèles doivent être efficaces et légers pour fonctionner dans les limites de ces dispositifs tout en fournissant des prédictions ou recommandations précises et opportunes.

Considérations éthiques : Les considérations éthiques, notamment la vie privée et la sécurité, sont essentielles lors de l'application de l'apprentissage machine dans la gestion du budget. Les données financières personnelles sont sensibles et doivent être traitées de manière sécurisée. De plus, il est crucial d'adresser le potentiel de biais dans les données ou les algorithmes pour assurer l'équité et prévenir la discrimination dans les systèmes d'apprentissage machine liés à la gestion du budget.

Engagement et adoption des utilisateurs : Un autre défi réside dans l'engagement et l'adoption des utilisateurs. Il est essentiel de concevoir des applications d'apprentissage machine de manière conviviale et intuitive pour encourager les individus à adopter et à utiliser la technologie. Une communication claire des avantages, des risques et des limitations des applications basées sur l'apprentissage machine pour la gestion du budget est cruciale pour favoriser la confiance des utilisateurs et un engagement à long terme.

En abordant ces défis et limitations, l'application de l'apprentissage machine dans la gestion du budget peut permettre des recommandations personnalisées, des plans budgétaires optimisés et des améliorations globales dans la gestion financière.

3.7 Conclusion

Dans ce chapitre, nous avons exploré les fondamentaux de l'apprentissage machine et son application dans le domaine de la gestion du budget.

Nous avons discuté de la pertinence de l'apprentissage supervisé, non supervisé et par renforcement pour optimiser la gestion budgétaire, améliorer les prédictions et personnaliser les plans budgétaires.

Nous avons également examiné les métriques d'évaluation pour évaluer la performance des modèles d'apprentissage machine liés à la gestion du budget, et souligné les défis et limitations de l'application de l'apprentissage machine dans ce domaine.

Dans les chapitres à venir, nous approfondirons des aspects spécifiques de l'apprentissage machine et son intégration avec la connaissance contextuelle pour améliorer encore les applications de gestion budgétaire.

Chapitre 4 méthodologie

4.1 Introduction à la Méthodologie

Ce chapitre fournit un aperçu des méthodes de développement utilisées pour la création de l'application Budget App. Les diagrammes du langage de modélisation unifié (UML) et leur importance seront discutés en profondeur, ainsi que les fonctionnalités et la structure de Budget App.

4.2 Langage de Modélisation Unifié (UML) et Ses Diagrammes

Le langage de modélisation unifié (UML) est un langage de modélisation visuel standardisé, utilisé de manière extensive en ingénierie logicielle pour la représentation et la conception des systèmes logiciels [23]. Cette section expliquera la définition et l'application de l'UML et de sa gamme de diagrammes.

4.2.1 Comprendre l'UML

L'UML fournit un langage graphique qui aide à capturer la structure, le comportement et les interactions des composants d'un système logiciel. Il sert d'outil de communication essentiel entre les développeurs de logiciels, les concepteurs et les parties prenantes, les aidant à comprendre l'architecture et la fonctionnalité du système [23].

4.2.2 Aperçu des Diagrammes UML

Les diagrammes UML peuvent être divisés en deux grandes catégories : les diagrammes structurels et les diagrammes comportementaux.

Diagrammes Structurels : Ils illustrent la structure statique des composants du système, en mettant en évidence les éléments et leurs relations [23]. Ils incluent des diagrammes tels que :

- **Diagramme de Classes** : Décrit les classes du système et les relations entre elles.
- **Diagramme de Composants** : Montre l'organisation et les dépendances entre un ensemble de composants.

- **Diagramme de Déploiement** : Représente l'architecture physique du système, incluant les nœuds matériels et les composants logiciels déployés sur ceux-ci.

Diagrammes Comportementaux : Ils décrivent le comportement dynamique et les interactions du système [23]. Parmi eux, on trouve :

- **Diagramme de Cas d'Utilisation** : Capture les interactions entre les acteurs externes et le système pour accomplir un objectif spécifique.
- **Diagramme de Séquence** : Représente les interactions entre objets sous forme de messages échangés dans une séquence temporelle.
- **Diagramme d'Activité** : Montre le flux de contrôle ou d'objets entre diverses activités d'un système.

4.3 Diagrammes UML clés dans le développement de l'application budget app

Dans le processus de développement de Budget app, plusieurs diagrammes UML ont joué un rôle déterminant :

- **Diagrammes de Cas d'Utilisation** : Ces diagrammes fournissent une vue orientée utilisateur des fonctionnalités du système, en identifiant les acteurs externes (utilisateurs ou systèmes externes) et les cas d'utilisation (actions ou services fournis par le système). Ils sont utiles pour définir les exigences fonctionnelles et clarifier les interactions entre les utilisateurs et le système [24] ;
- **Diagrammes de Séquence** : Ces diagrammes montrent la séquence d'événements et d'interactions entre les objets du système pour réaliser un comportement spécifique. Ils mettent en évidence les messages échangés entre les objets au fil du temps, offrant une vue détaillée de la logique opérationnelle du système [24] ;
- **Diagrammes de Classes** : Ces diagrammes capturent la structure statique du système en identifiant les classes, leurs attributs, opérations et relations. Ils sont utilisés pour

modéliser l'architecture du système en spécifiant les entités de base (classes) et en décrivant leurs interactions et hiérarchies [24].

4.4 c'est quoi budget app ?

L'application mobile innovante "Budget App" vise à autonomiser ses utilisateurs dans la gestion efficace de leurs finances personnelles, en leur offrant une gamme complète de fonctionnalités clés :

4.4.1 Budget App fonctionnalité

1. Page d'Accueil

Description : La page d'accueil est la première vue lorsque l'utilisateur se connecte à l'application. Elle offre une vue d'ensemble des finances personnelles à travers un graphique circulaire des dépenses et des revenus, ainsi qu'une liste détaillée des dépenses récentes.

- **Graphique Circulaire des Dépenses et Revenus :**
 - **Fonction :** Fournit une représentation visuelle proportionnelle des dépenses et des revenus.
 - **Interaction :** L'utilisateur peut voir instantanément la répartition de ses finances entre dépenses et revenus.
- **Liste des Dépenses**
 - **Fonction :** Affiche les détails des dépenses récentes, y compris la date, le montant, la catégorie et la description.
 - **Interaction :** Permet à l'utilisateur de visualiser toutes les dépenses en un seul endroit pour une gestion facile.
- **Filtrage par Date et Type :**
 - **Fonction :** Permet à l'utilisateur de filtrer les dépenses et revenus par date spécifique (jour, semaine, mois, année) et par type (dépense ou revenu).
 - **Interaction :** Facilite l'analyse des habitudes de dépenses et de revenus sur différentes périodes et catégories.

- **Ajout de Dépenses ou Revenus :**

- **Fonction :** Intègre un bouton d'ajout pour faciliter l'entrée rapide de nouvelles dépenses ou revenus.
- **Interaction :** En cliquant sur ce bouton, l'utilisateur peut remplir un formulaire avec la date, le montant, la catégorie et une description, ajoutant ainsi une nouvelle entrée à sa gestion financière.

- **Reconnaissance Vocale**

- **Fonction :** Intègre une fonctionnalité avancée de reconnaissance vocale pour permettre l'ajout automatique de dépenses ou revenus par commande vocale.
- **Interaction :** Lorsque l'utilisateur active la fonction de reconnaissance vocale en appuyant sur le bouton IA, il peut dicter verbalement les détails de la transaction financière.
- **Traitement Intelligent :** Les commandes vocales sont analysées par des algorithmes d'intelligence artificielle embarqués dans l'application. Ces algorithmes comprennent et extraient automatiquement les informations essentielles telles que la date, le montant, la catégorie et la description de la transaction.

- **Confirmation et Enregistrement :** Après avoir interprété avec précision la commande vocale, l'application affiche une confirmation à l'utilisateur pour s'assurer de l'exactitude des détails. Une fois confirmée par l'utilisateur, la transaction est automatiquement enregistrée dans la base de données financière de l'application.
- **Avantages :** Cette fonctionnalité améliore significativement l'efficacité en permettant aux utilisateurs de mettre à jour leurs finances personnelles rapidement et sans effort, tout en minimisant les erreurs de saisie manuelle.

2. Graphiques

Description : Cette fonctionnalité permet à l'utilisateur d'explorer ses données financières à travers un graphique en barres interactif.

- **Graphique en Barres :**

- **Fonction** : Offre une visualisation détaillée des dépenses et des revenus selon différents critères de filtrage : général, revenus ou dépenses.
- **Interaction** : L'utilisateur peut sélectionner une période spécifique (jour, semaine, mois, année) ou un type de données (revenus ou dépenses) pour analyser ses finances plus en profondeur.
- **Filtrage** :
 - **Fonction** : Inclus des options de filtrage pour afficher les données générales, les revenus seulement ou les dépenses seulement dans le graphique en barres.
 - **Interaction** : Permet à l'utilisateur de comparer et de contraster ses habitudes de dépenses et de revenus selon différents paramètres sélectionnés.

3. Budgets

Description : Cette fonctionnalité permet à l'utilisateur de définir et de gérer ses budgets mensuels.

- **Liste des Budgets** :
 - **Fonction** : Affiche tous les budgets mensuels définis par l'utilisateur, comprenant la catégorie, le montant fixe alloué, la description et l'état actuel de l'utilisation des fonds.
 - **Interaction** : L'utilisateur peut consulter et ajuster ses budgets en fonction de ses objectifs financiers mensuels.
- **Ajout et Mise à Jour de Budgets** :
 - **Fonction** : Intègre des fonctionnalités pour ajouter de nouveaux budgets et mettre à jour les budgets existants.
 - **Interaction** : En cliquant sur le bouton d'ajout ou en sélectionnant un budget existant, l'utilisateur peut définir ou ajuster les paramètres du budget, tel que le montant alloué et la description.

4. Dettes et Crédits

Description : Affiche une vue des dettes et crédits en cours, permettant à l'utilisateur de gérer ses obligations financières.

- **Liste des Dettes et Crédits :**

- **Fonction :** Présente toutes les obligations financières actives, avec des détails tels que le montant dû, la date d'échéance et le type (dette ou crédit).
- **Interaction :** L'utilisateur peut consulter toutes ses obligations financières en un seul endroit pour un suivi efficace.

- **Ajout et Mise à Jour de Dettes/Crédits :**

- **Fonction :** Inclus des fonctionnalités pour ajouter de nouvelles dettes ou crédits, ainsi que pour mettre à jour les montants existants.
- **Interaction :** En sélectionnant une entrée de dette ou de crédit, l'utilisateur peut entrer de nouvelles informations ou ajuster les montants existants selon les paiements effectués ou les nouvelles transactions.

5. Objectif

Description : Cette fonctionnalité permet aux utilisateurs de définir, suivre et gérer leurs objectifs financiers personnels, offrant une vision claire de leurs aspirations financières à atteindre.

- **Liste des Objectifs :**

- **Fonction :** Affiche tous les objectifs financiers actuels de l'utilisateur, répertoriant chaque objectif avec sa catégorie, le montant cible, la date limite et une description.
- **Interaction :** L'utilisateur peut consulter la liste complète de ses objectifs financiers depuis l'écran principal de l'application. Chaque objectif est présenté de manière organisée pour permettre un suivi précis de leurs progrès.

- **Ajout d'Objectifs :**

- **Fonction :** Permet à l'utilisateur d'ajouter de nouveaux objectifs financiers à sa liste existante.
- **Interaction :** En sélectionnant l'option d'ajout d'objectif, l'utilisateur peut spécifier La catégorie de l'objectif (par exemple, épargne, investissement, réduction de dettes), le montant cible à atteindre, la date limite pour l'objectif et une description pour clarifier l'objectif.

- **Modification d'Objectifs :**

- **Fonction** : Offre la possibilité de mettre à jour les détails des objectifs financiers existants.
- **Interaction** : En sélectionnant un objectif spécifique dans la liste, l'utilisateur peut Modifier le montant cible si nécessaire.
- **Suivi des Progrès** :
 - **Fonction** : Fournit des indicateurs visuels ou numériques pour suivre les progrès réalisés vers chaque objectif.
 - **Interaction** : L'utilisateur peut voir facilement les progrès accomplis vers ses objectifs financiers grâce à des graphiques ou des pourcentages affichés dans l'application. Cela permet de maintenir la motivation et de prendre des décisions financières éclairées.

4.4.2 Diagramme de cas d'utilisation

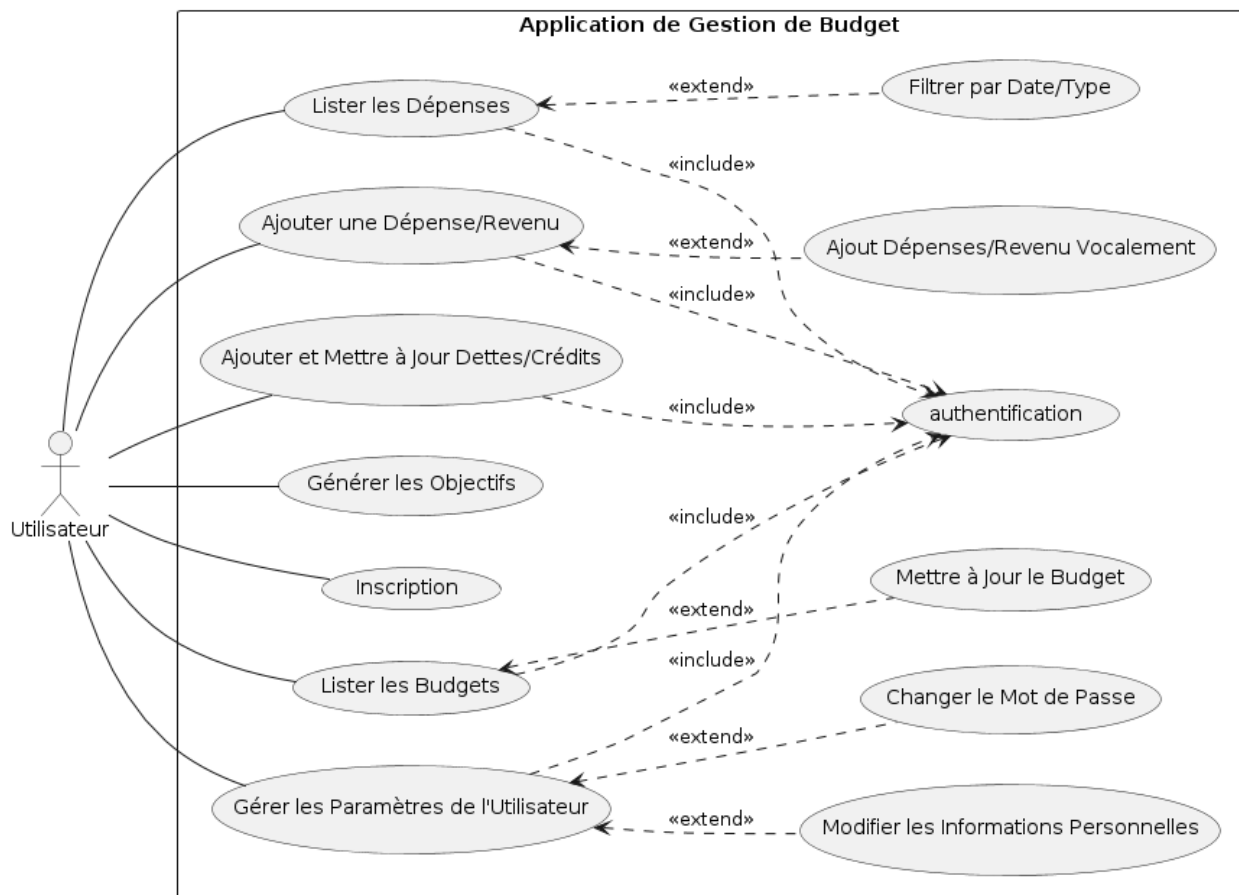


Figure 10 : Diagramme de cas d'utilisation de la conception Budget App

La description textuelle du diagramme cas d'utilisation est la suivante :

- **Cas d'utilisation : Inscription**

Acteur : Utilisateur

Précondition : L'utilisateur doit être sur la page d'inscription et avoir une adresse e-mail valide.

Postcondition : L'utilisateur est inscrit avec succès et peut se connecter à l'application.

Scénario Nominal

1. L'utilisateur accède à la page d'inscription de l'application.
2. Le système affiche le formulaire d'inscription avec les champs requis (email, mot de passe, nom, etc.).
3. L'utilisateur remplit le formulaire et valide.
4. Le système vérifie la validité des informations saisies.
5. Le système vérifie si l'e-mail n'est pas déjà enregistré.
6. Le système crée un nouveau compte utilisateur.
7. L'utilisateur peut maintenant se connecter.

Scénario Alternatif

4.a Le système affiche une erreur "Le mot de passe et la confirmation du mot de passe ne correspondent pas".

- Retour à l'étape 3.

5.a Le système affiche une erreur "L'e-mail est déjà enregistré".

- Retour à l'étape 3.

Scénario d'Erreur

3.e. L'utilisateur entre une adresse e-mail invalide.

- Le système affiche une erreur correspondante et bloque la soumission du formulaire.

3.e. L'utilisateur entre un mot de passe qui ne répond pas aux exigences de sécurité.

- Le système affiche une erreur correspondante et bloque la soumission du formulaire.

3.e. L'utilisateur entre une confirmation de mot de passe qui ne correspond pas au mot de passe saisi.

- Le système affiche une erreur correspondante et bloque la soumission du formulaire.

- Lister les Dépenses

Acteur : Utilisateur

Précondition : L'utilisateur doit être authentifié.

Postcondition : L'utilisateur voit la liste des dépenses filtrée par date ou type si un filtre est appliqué.

Scénario nominal :

1. L'utilisateur ouvre l'application.
2. L'utilisateur se connecte avec succès.
3. L'utilisateur navigue vers la section "Lister les Dépenses".
4. Le système affiche toutes les dépenses de l'utilisateur.
5. L'utilisateur applique un filtre par date ou type.
6. Le système affiche les dépenses correspondant aux critères de filtre.

Scénario alternatif :

- a.5. L'utilisateur choisit de ne pas appliquer de filtre.
- a.6. Le système affiche toutes les dépenses sans filtre.

Scénario d'erreur :

e.2. L'utilisateur échoue à se connecter (mauvais identifiants).

e.3. Le système affiche un message d'erreur indiquant que les identifiants sont incorrects.

- **Cas d'utilisation : reconnaissance vocale**

Acteur : Utilisateur

Précondition : L'utilisateur doit cliquer sur le Button IA.

Postcondition : l'utilisateur a accepté les permissions vocales

Scénario Nominal

1. L'utilisateur click sur le Button IA.
2. L'application commence à écouter.
3. L'application envoie le vocal au système pour le traiter et extraire les mots clés.
4. Après le traitement le système envoie les mots clés
5. L'application ajoute automatiquement les dépenses ou revenus

Scénario Alternatif

3.a le système affiche une erreur " quelque chose a interrompu l'écoute. "

4.a le système affiche une erreur « erreur les mots clés ne correspondent pas. "

Scénario d'Erreur

3.e l'utilisateur parle avec une autre langue

- Le système affiche une erreur correspondante

3.e l'utilisateur donne des mots inconnus

- Le système affiche une erreur correspondante

Ajouter une Dépense/Revenu

Acteur : Utilisateur

Précondition : L'utilisateur doit être authentifié.

Postcondition : Une nouvelle dépense ou un nouveau revenu est ajouté à la liste des transactions de l'utilisateur.

Scénario nominal :

1. L'utilisateur ouvre l'application.
2. L'utilisateur se connecte avec succès.
3. L'utilisateur navigue vers la section "Ajouter une Dépense/Revenu".
4. L'utilisateur saisit les détails de la dépense ou du revenu.
5. L'utilisateur soumet le formulaire.
6. Le système enregistre la nouvelle dépense ou le nouveau revenu et met à jour la liste des transactions.

Scénario alternatif :

- a.4. L'utilisateur choisit d'ajouter une dépense/revenu vocalement.
- a.5. L'utilisateur utilise la commande vocale pour entrer les détails.
- a.6. Le système reconnaît et enregistre la dépense ou le revenu et met à jour la liste des transactions.

Scénario d'erreur :

- e.2. L'utilisateur échoue à se connecter (mauvais identifiants).
- e.3. Le système affiche un message d'erreur indiquant que les identifiants sont incorrects.
- e.5. Le formulaire de saisie est incomplet ou incorrect.

e.6. Le système affiche un message d'erreur demandant à l'utilisateur de corriger les informations.

- **Cas d'utilisation : Ajouter et Mettre à Jour Dettes/Crédits**

Acteur : Utilisateur

Précondition : L'utilisateur doit être connecté à l'application.

Postcondition : Une nouvelle dette ou un nouveau crédit est ajouté

Scénario Nominal

1. L'utilisateur sélectionne l'option pour voir les dettes et crédits.
2. Le système affiche la liste des dettes et crédits actuels.
3. L'utilisateur peut voir les détails de chaque dette ou crédit.

Scénario Alternatif

- Aucun scénario alternatif prévu pour cette fonctionnalité.

Scénario d'Erreur

- Aucun scénario d'erreur prévu pour cette fonctionnalité

- **Cas d'utilisation : générer les Objectifs**

Acteur : Utilisateur

Précondition : L'utilisateur doit être connecté à l'application.

Postcondition : Un nouvel objectif est ajouté à la liste des objectifs existants, ou un objectif existant est mis à jour avec les nouvelles informations

Scénario Nominal

1. L'utilisateur sélectionne l'option "Objectifs" dans le menu de l'application.
2. Le système affiche la liste actuelle des objectifs de l'utilisateur.
3. L'utilisateur ajout un nouvel objectif.
4. L'utilisateur mets à jour d'un objectif existant

Scénario Alternatif

3.a l'utilisateur annule l'action d'ajout ou de mise à jour à n'importe quelle étape, le système retourne à l'état précédent sans effectuer de modifications.

Scénario d'Erreur

3.e les données saisies dans le formulaire ne sont pas, le système affiche un message d'erreur spécifiant les corrections nécessaires.

4.4.3 Diagramme de séquence

- Diagramme de séquence de l'inscription

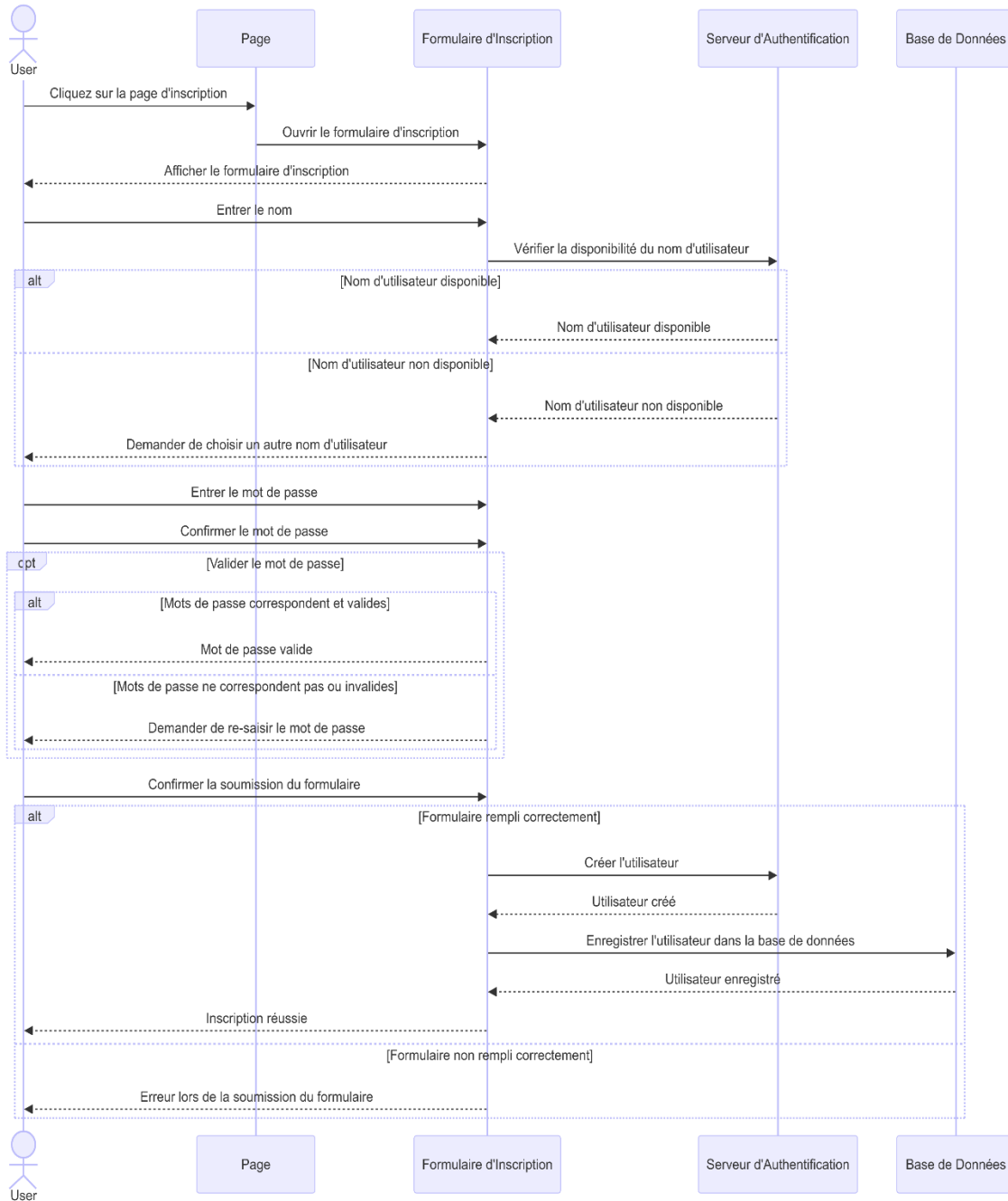


Figure 11 : Diagramme de séquence représentant la phase d'inscription

Explication :

Ce diagramme de séquence montre le processus d'inscription d'un utilisateur dans l'application de gestion de budget. L'utilisateur accède à la page d'inscription, remplit le formulaire avec ses informations personnelles, et le système vérifie la validité de ces informations. Si tout est correct, le système crée un nouveau compte utilisateur et confirme l'inscription.

- **Diagramme de séquence de l'Accueil**

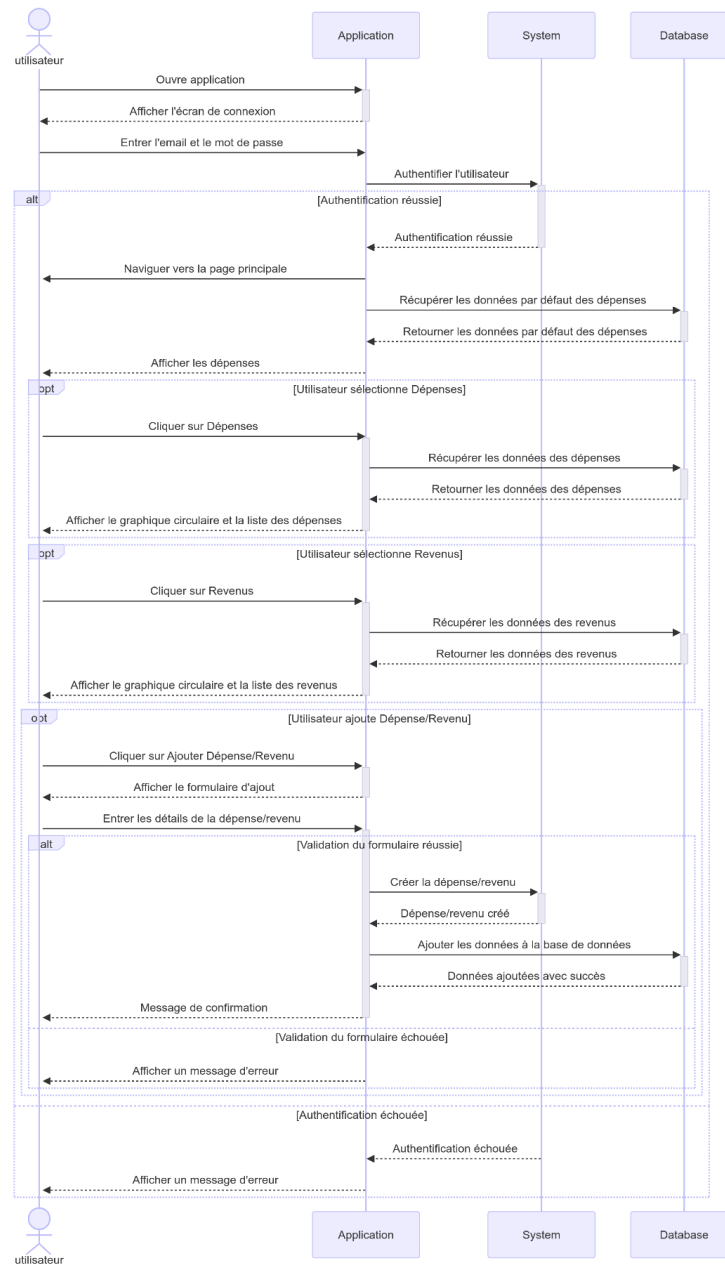


Figure 12 : Diagramme de séquence représentant la page d'accueil

Explication :

Ce diagramme de séquence montre comment l'utilisateur interagit avec la page d'accueil de l'application. L'utilisateur ouvre l'application, et celle-ci affiche un diagramme circulaire des dépenses et des revenus ainsi qu'une liste des dépenses récentes. L'utilisateur peut ensuite appliquer des filtres ou ajouter de nouvelles dépenses ou revenus.

- **Diagramme de séquence de Reconnaissance vocal**

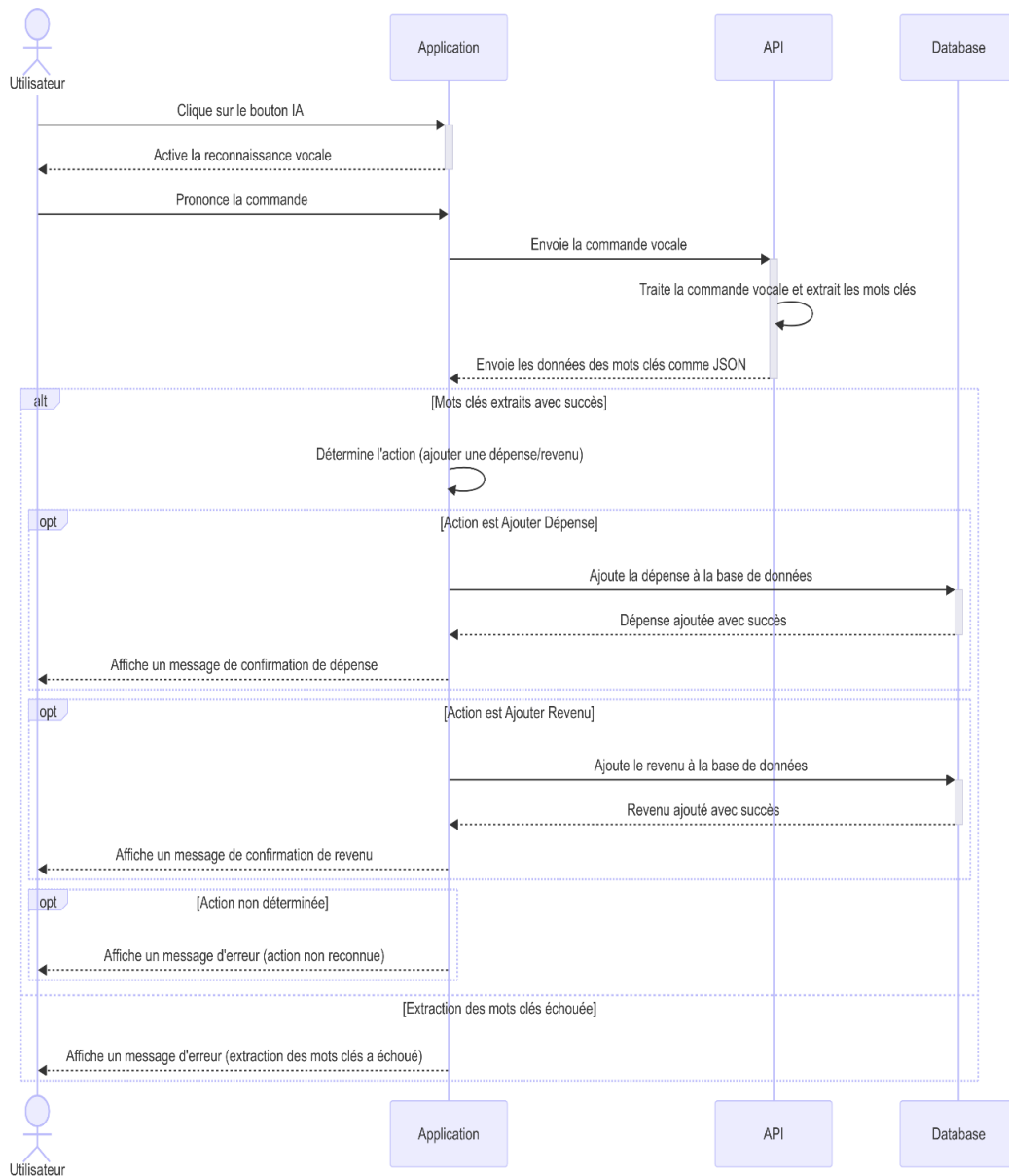


Figure 13 : Diagramme de séquence représentant le Button IA

Explication :

Ce diagramme de séquence montre comment l'utilisateur utilise la reconnaissance vocale pour ajouter des dépenses ou des revenus automatiquement dans l'application. L'utilisateur active la reconnaissance vocale, dicte les détails de la dépense ou du revenu, et le système enregistre les informations après confirmation.

- **Diagramme de séquence de la page Graphique**

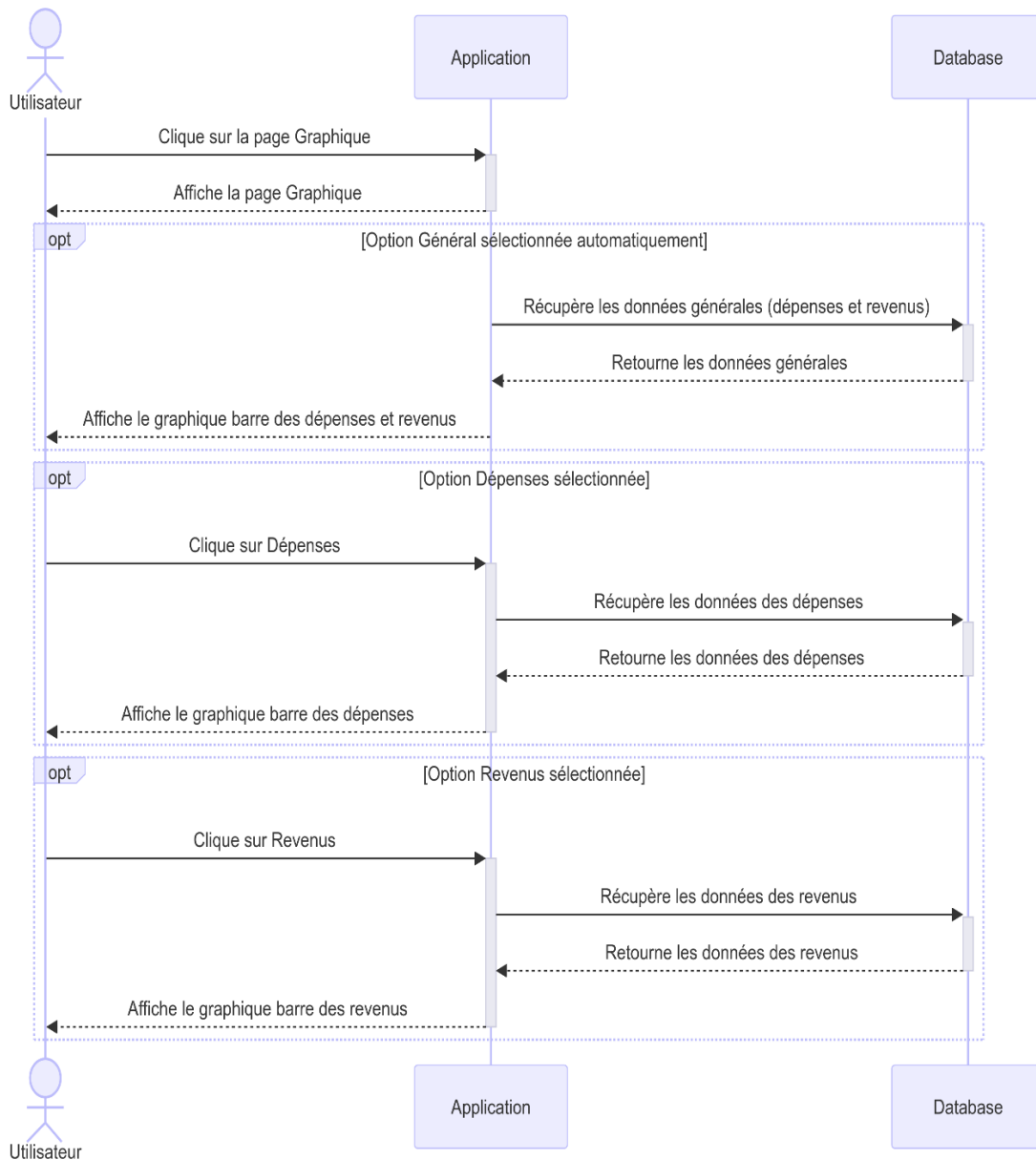


Figure 14 : Diagramme de séquence représentant la page Graphique

Explication :

Ce diagramme de séquence montre comment l'utilisateur interagit avec la fonctionnalité des graphiques de l'application. L'utilisateur accède à la section des graphiques, et l'application affiche un diagramme en barres. L'utilisateur peut ensuite filtrer les données par général, revenus ou dépenses.

- **Diagramme de séquence de la page Dettes/Crédits**

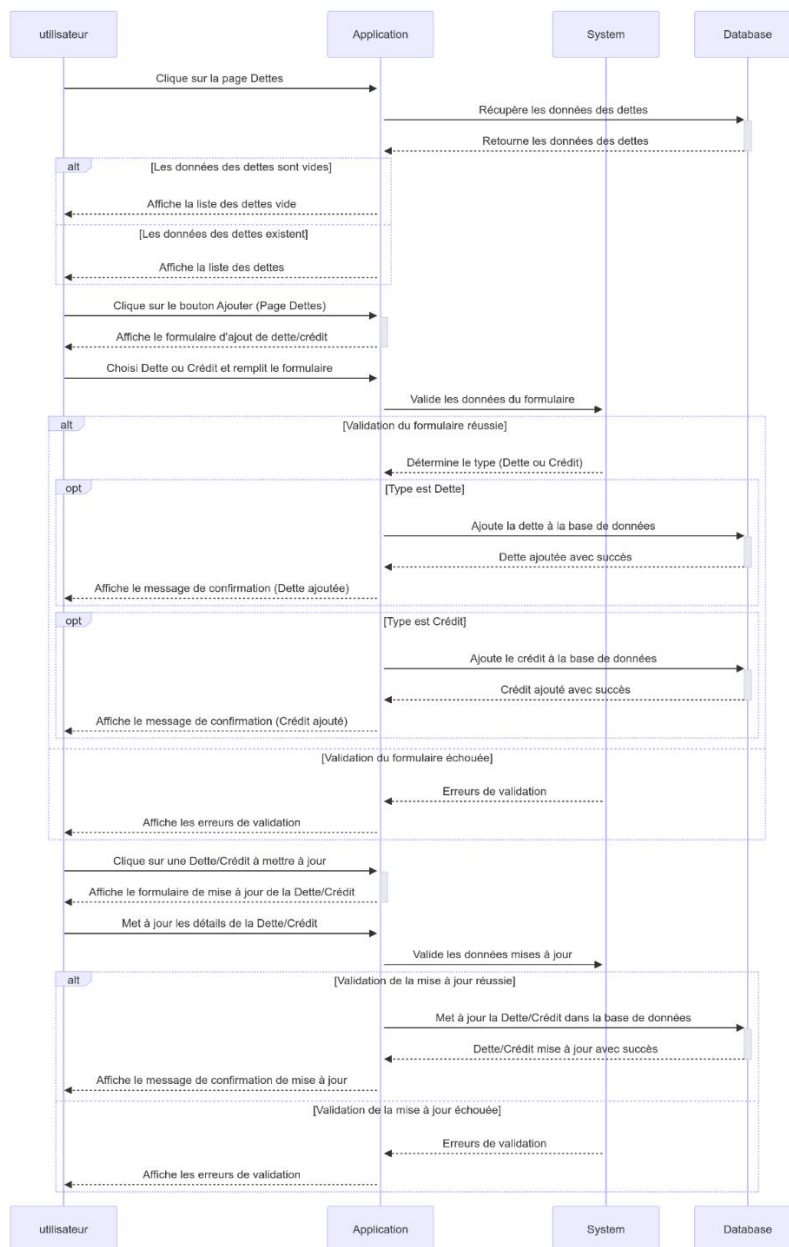


Figure 15 : Diagramme de séquence représentant la page Dettes/Crédits

Explication :

Ce diagramme de séquence montre comment l'utilisateur interagit avec la fonctionnalité de gestion des dettes et crédits de l'application. L'utilisateur accède à la section des dettes, et l'application affiche une liste des dettes et crédits actuels. L'utilisateur peut ajouter une nouvelle dette ou crédit ou mettre à jour une dette ou un crédit existant

- **Diagramme de séquence de la page Budgets**

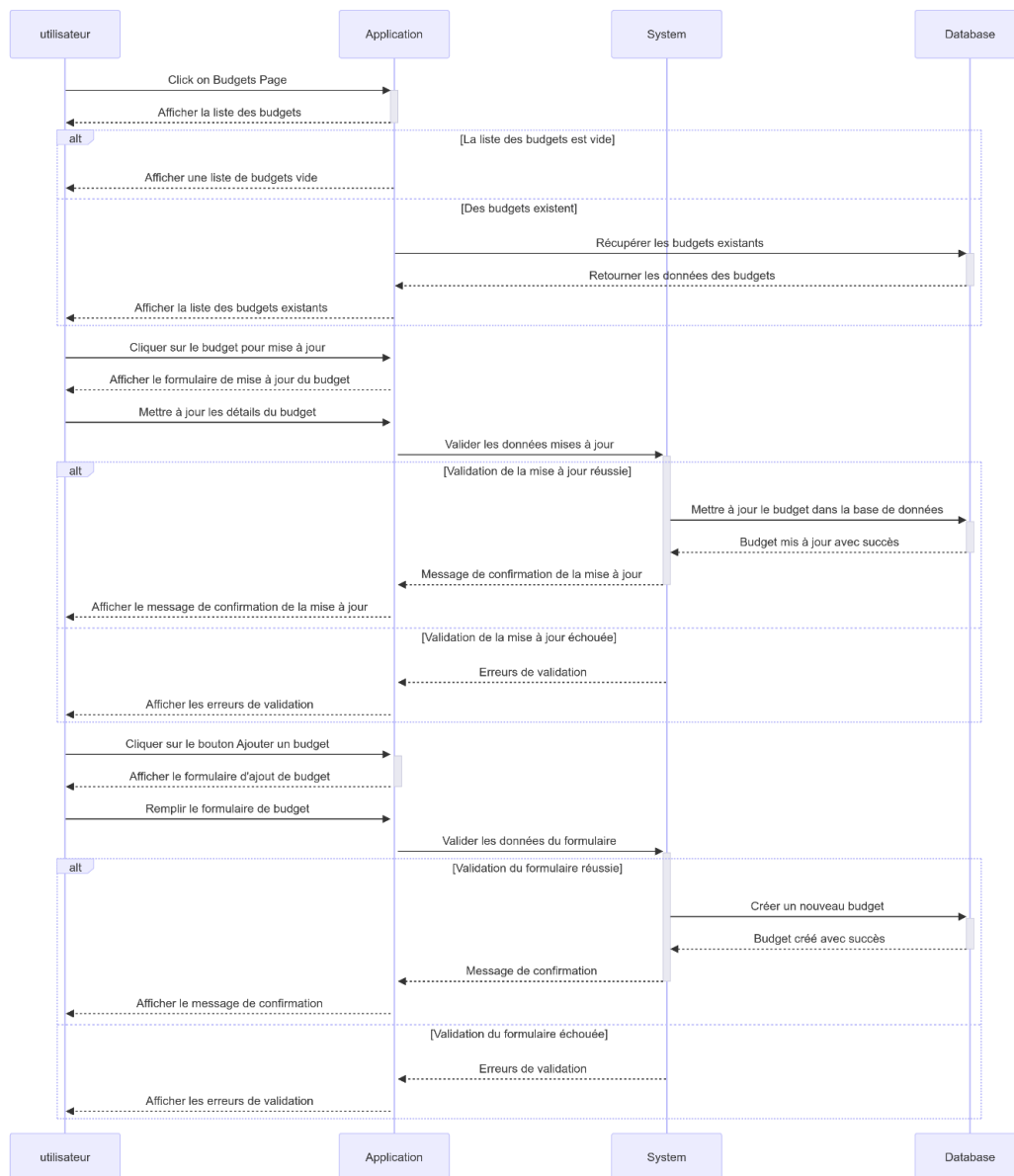


Figure 16 : Diagramme de séquence représentant les Budgets

Explication :

Ce diagramme de séquence montre comment l'utilisateur interagit avec la fonctionnalité de gestion des budgets de l'application. L'utilisateur accède à la section des budgets, et l'application affiche une liste des budgets actuels. L'utilisateur peut ajouter un nouveau budget ou mettre à jour un budget existant.

- **Diagramme de séquence de la page Objectifs**

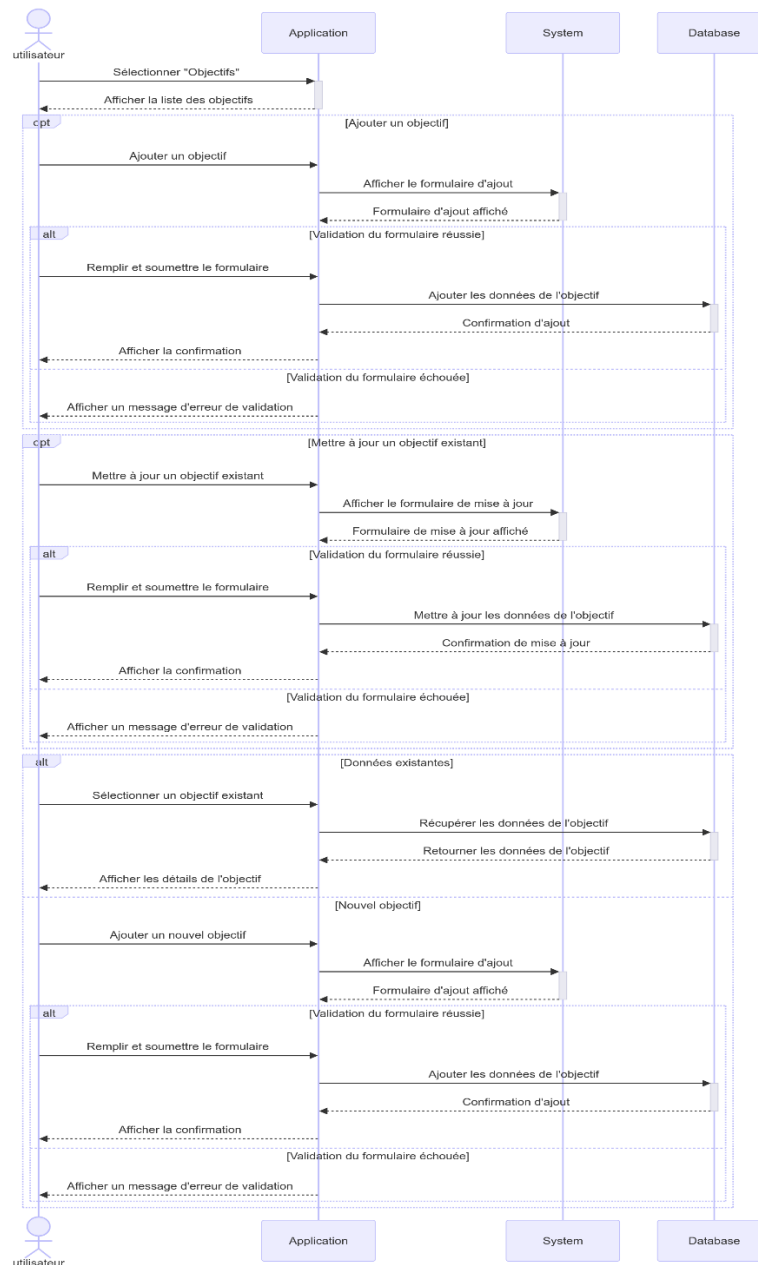


Figure 17 : Diagramme de séquence représentant les objectifs

Explication :

Ce diagramme de séquence montre comment l'utilisateur interagit avec la fonctionnalité de gestion des objectifs financiers de l'application. L'utilisateur accède à la section des objectifs, et l'application affiche une liste des objectifs actuels. L'utilisateur peut ajouter un nouvel objectif ou mettre à jour un objectif existant.

4.4.4 Diagramme de classe

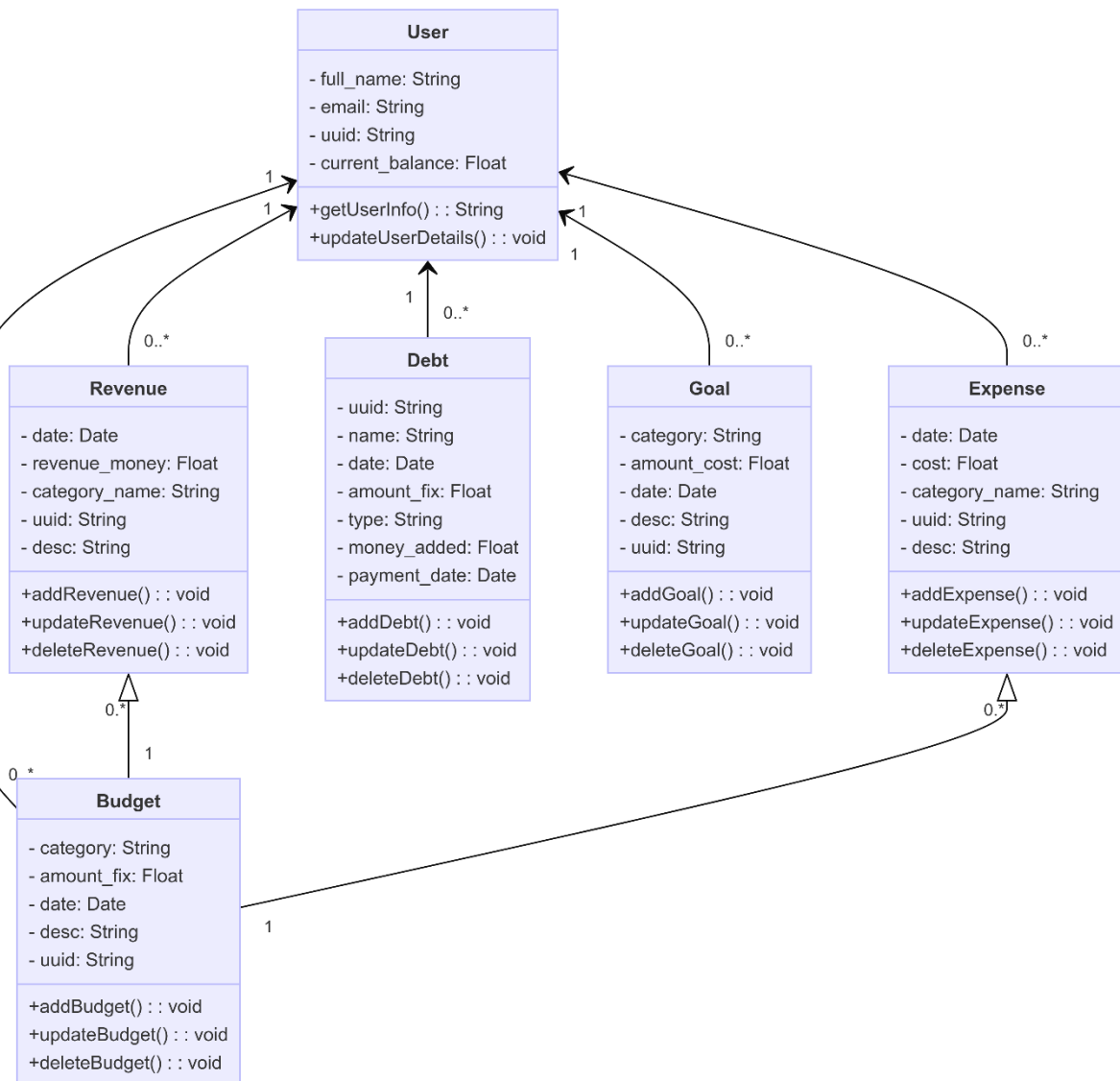


Figure 18 : Diagramme de classe de la conception Budget App

4.5 Modèle Proposé

4.5.1 Introduction

L'objectif de notre modèle de formation est de prédire le temps nécessaire pour atteindre un objectif d'économies pour des catégories telles que les voyages et les voitures. De plus, le modèle recommande des sous-catégories basées sur le montant de l'épargne, visant à aider les utilisateurs à prendre des décisions financières éclairées.

4.5.2 Collecte et prétraitement des données

Les données ont été créées manuellement, à partir de divers scénarios, notamment les dossiers financiers. Il a ensuite été nettoyé pour gérer les valeurs manquantes et les valeurs aberrantes, et les caractéristiques ont été normalisées pour garantir l'uniformité. Les variables catégorielles ont été codées de manière appropriée [25].

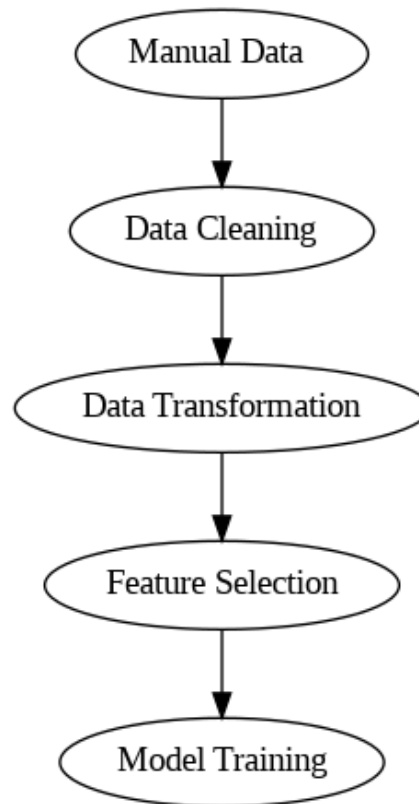


Figure 19 : Collecte et prétraitement des données

4.5.4 Sélection de fonctionnalité

Nos caractéristiques d'entrée comprenaient le montant des économies et la catégorie choisie, tandis que les variables cibles étaient le temps nécessaire pour réaliser les économies et les sous-catégories recommandées [25].

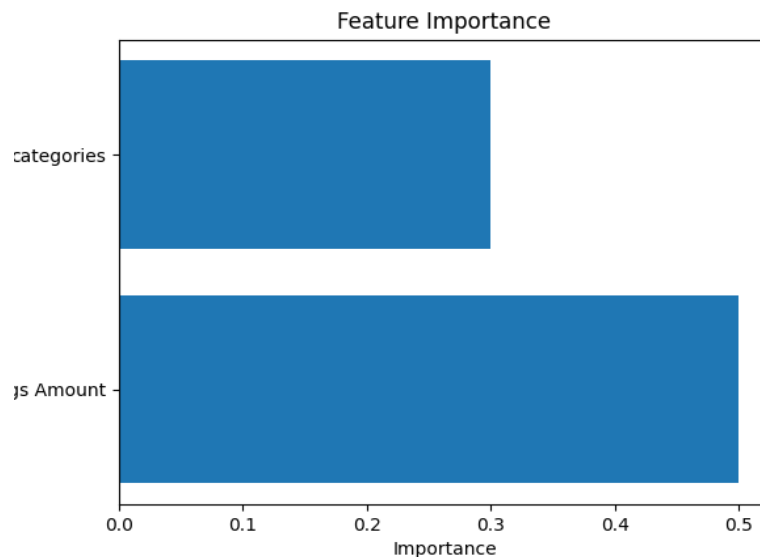


Figure 20 : Feature importance and importance

4.5.5 Architecture du modèle

Nous avons utilisé un modèle de réseau neuronal avec plusieurs couches denses. La fonction d'activation ReLU était utilisée dans des couches cachées, tandis que la couche de sortie utilisait une fonction d'activation linéaire. La fonction de perte d'erreur quadratique moyenne et l'optimiseur Adam ont été choisis pour la formation [25].

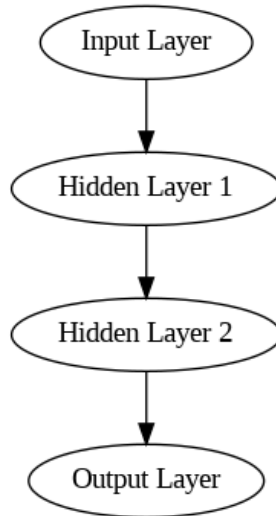


Figure 21 : Architecture du modèle

4.5.6 Processus de formation

Les données ont été réparties en 80 % d'ensembles de formation et 20 % de validation. Les hyperparamètres tels que le taux d'apprentissage et la taille des lots ont été ajustés à l'aide de la recherche par grille. Le modèle a été formé pendant 50 époques avec une taille de lot de 32, en utilisant un GPU de google collab afin d'accélérer le processus [25].

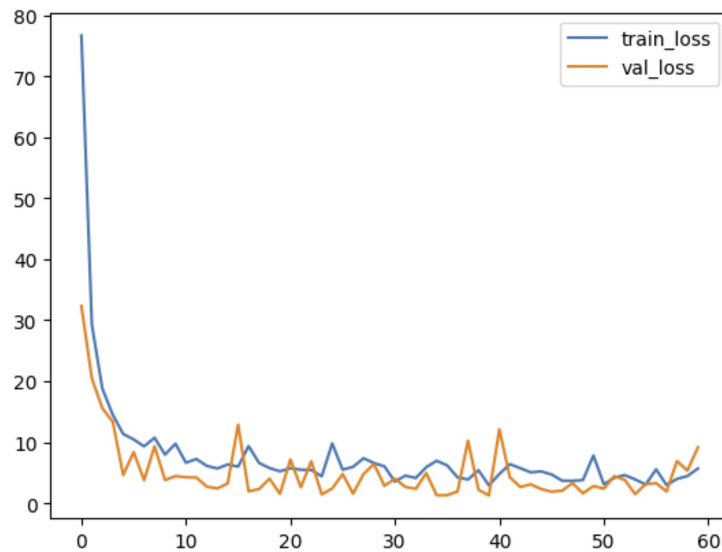


Figure 22 : train_loss and val_loss

4.5.7 Paramètres d'évaluation

Les performances du modèle ont été évaluées à l'aide de l'erreur absolue moyenne et de l'erreur quadratique moyenne. Les résultats de validation ont montré une précision assez prometteuse, avec des graphiques pertinents indiquant les performances du modèle au fil des époques [25].

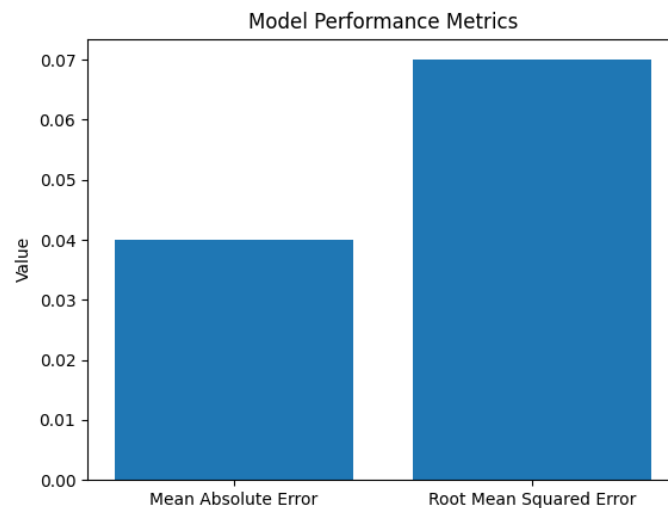


Figure 23 : Mean absolute error/ squared error

4.5.8 Conversion du modèle

Le modèle formé a été converti de TensorFlow en TensorFlow Lite pour faciliter le déploiement sur les appareils mobiles. Des techniques telles que la quantification a été appliquées pour optimiser le modèle pour les environnements aux ressources limitées [26].

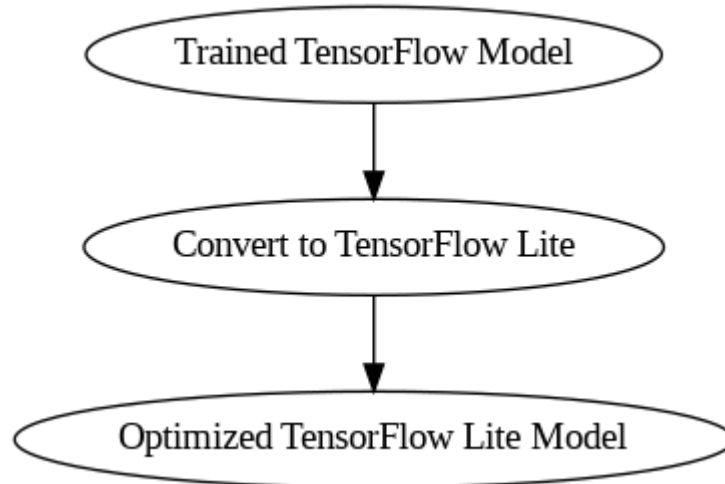


Figure 24 : conversion du modèle

4.5.9 Déploiement et tests

Le modèle a été déployé dans une application mobile utilisant TensorFlow Lite. Des tests approfondis ont été menés pour garantir la précision et l'efficacité du modèle dans des scénarios réels, en relevant les défis liés au déploiement.

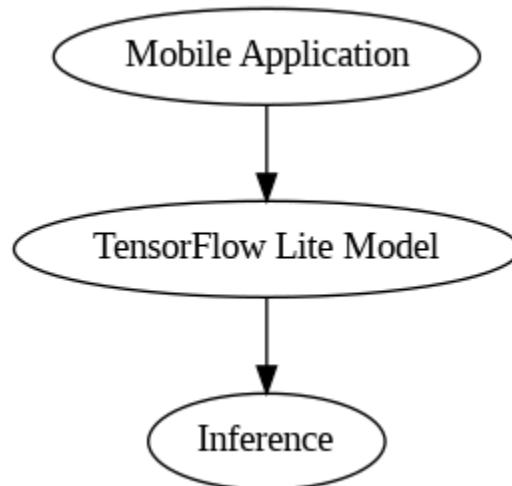


Figure 25 : déploiement et tests

4.6 Conclusion

Dans ce chapitre nous avons discuté des méthodologies utilisées pour développer l'application Budget App, en mettant en avant l'importance des diagrammes UML dans le développement logiciel. Nous avons présenté les fonctionnalités principales et la structure de l'application Budget App à travers différents diagrammes UML.

Après avoir établi les bases théoriques et méthodologiques, nous passerons au chapitre suivant, où nous discuterons du processus de mise en œuvre réel et de la pile technologique utilisée pour construire Budget App.

Chapitre 5 : implémentation

5.1 Introduction

Dans ce chapitre, nous aborderons le processus d'implémentation de l'application de gestion de budget, en nous concentrant sur l'architecture, langage de programmation, Framework utilisées et les outils choisie pour son développement.

5.2 Architecture Model View ViewModel (MVVM)

5.2.1 Modèle (Model)

Le modèle représente la couche de données de l'application. Il inclut les structures de données, les logiques métiers, les services d'accès aux données et les API. Dans notre application de gestion de budget, le modèle gère les entités telles que les dépenses, les revenus, les budgets, les dettes et les objectifs. Il est responsable de la récupération, de la mise à jour et de la gestion des données [27].

5.2.2 Vue (View)

La vue est responsable de l'affichage des données et de l'interface utilisateur. Elle ne contient aucune logique métier et se contente de représenter visuellement les informations fournies par le ViewModel. Dans l'application, la vue comprend les formulaires d'ajout de dépenses, les listes de budgets, les graphiques des revenus et des dépenses, etc. La vue est mise à jour automatiquement grâce à la liaison de données avec le ViewModel [27].

5.2.3 Vue Modèle (ViewModel)

Le ViewModel agit comme un intermédiaire entre le modèle et la vue. Il contient la logique de présentation et gère les états de l'interface utilisateur. Le ViewModel expose des données et des commandes que la vue peut lier. Il surveille les modifications dans le modèle et met à jour la vue en conséquence. Le ViewModel n'a pas de connaissance directe de la vue, ce qui permet de tester la logique de présentation de manière isolée [27].

5.2.4 Interaction entre les composants

- **Model** informe le ViewModel des modifications de données, permettant à la vue de se mettre à jour automatiquement.
- **View** observe les propriétés et les commandes du ViewModel via des liaisons de données.
- **Le ViewModel** interagit avec le Modèle pour récupérer et manipuler les données nécessaires.

5.2.5 Avantages de l'Architecture MVVM

- **Séparation des préoccupations** : Chaque composant a une responsabilité bien définie, facilitant la maintenance du code.
- **Testabilité** : La logique de présentation étant séparée de la vue, elle peut être testée de manière indépendante.
- **Réutilisabilité** : Les composants du ViewModel peuvent être réutilisés dans différentes vues, et vice versa.
- **Gestion facile des états** : Le ViewModel simplifie la gestion des états de l'interface utilisateur, rendant l'application plus réactive et fluide.

5.3 Matériel utilisé

Le travail de développement a été réalisé à l'aide d'un ordinateur avec les spécifications suivantes :

- **Processeur** : 11th Gen Intel(R) Core (TM) i7-11800H @ 4.60 GHz ;
- **Mémoire** : 32.00 GB RAM de type DDR4-3200 ;
- **Stockage**: 512 M.2 2280 solid-state drive;
- **Carte Graphique** : NVIDIA GeForce RTX 3050 Ti ;
- **Système opérateur** : Microsoft Windows 11.

5.4 Technologie et Framework

Dans ce chapitre, nous examinerons en détail les technologies et les cadres spécifiques employés lors de la création de l'application Budget App.

Cela englobe différents langages de programmation, frameworks et bibliothèques, ainsi que les outils utilisés lors du développement.

5.4.1 Langage de programmation



Dart est un langage de programmation optimisé pour les applications client, développé par Google. Il est utilisé principalement avec Flutter pour le développement d'applications mobiles. Dart offre une syntaxe simple, un fort typage, et est compilé à la fois en code natif et en JavaScript pour une exécution rapide [28].



Python est un langage de programmation interprété, orienté objet et de haut niveau, avec une sémantique dynamique. Il est largement utilisé pour le développement de logiciels, l'analyse de données et les applications d'intelligence artificielle [29].

5.4.2 Frameworks et bibliothèques



Flutter est un Framework open-source développé par Google pour créer des applications nativement compilées pour mobile, web et desktop à partir d'une seule base de code. Il utilise le langage Dart et offre un riche ensemble de widgets personnalisables pour créer des interfaces utilisateur attrayantes et performantes [20].



Supabase est une plateforme open-source qui fournit des services backend tels que l'authentification, la base de données, le stockage et les fonctions serverless. Il est basé sur PostgreSQL et simplifie grandement le développement d'applications avec des fonctionnalités prêtes à l'emploi [31].



Wit.ai est une plateforme de traitement du langage naturel (NLP) qui permet aux développeurs d'intégrer des fonctionnalités de reconnaissance vocale et de compréhension du langage dans leurs applications. Elle est utilisée pour l'ajout automatique de dépenses et de revenus via des commandes vocales [32].



TensorFlow est une bibliothèque open-source de machine learning développée par Google. Elle est utilisée pour les applications d'intelligence artificielle, en particulier pour le deep learning [33].



TensorFlow Lite est une version légère de TensorFlow, conçue pour les appareils mobiles et embarqués. Il permet d'exécuter des modèles de machine learning avec des contraintes de ressources réduites [26].



cikit-learn est une bibliothèque open-source de machine learning pour le langage de programmation Python. Elle offre des outils simples et efficaces pour l'analyse et le data mining [34].



Keras est une bibliothèque open-source de réseaux de neurones écrite en Python. Elle est conçue pour permettre une expérimentation rapide avec des réseaux neuronaux profonds et est intégrée à TensorFlow [35].



NumPy est une bibliothèque pour le langage Python qui ajoute le support des tableaux et matrices multidimensionnels, ainsi que des fonctions mathématiques de haut niveau pour opérer sur ces tableaux [36].



Pandas est une bibliothèque open-source de manipulation et d'analyse de données pour Python. Elle offre des structures de données et des outils de manipulation de données de haut niveau [37].

5.4.3 Outils de développement



Visual Studio Code (VS Code) est un éditeur de code source développé par Microsoft pour Windows, Linux et macOS. Il inclut le support pour le débogage, le contrôle de version Git, des fonctionnalités d'édition de code telles que la complétion de code, des refactorisations, et la coloration syntaxique [38].



Git est un système de contrôle de version distribué utilisé pour suivre les modifications apportées au code source pendant le développement du logiciel. Il permet à plusieurs développeurs de travailler ensemble de manière efficace et sécurisée [39].



GitHub est une plateforme de développement qui permet de stocker et de gérer du code, ainsi que de suivre et contrôler les modifications apportées. Il fournit des fonctionnalités de collaboration et de versioning via Git [40].



Figma est un outil de conception d'interface utilisateur basé sur le web. Il permet aux concepteurs de collaborer en temps réel et offre des fonctionnalités pour la création de prototypes et de designs interactifs [41].



Postman est un outil de développement API qui permet de tester, documenter et automatiser les API. Il fournit une interface utilisateur conviviale pour envoyer des requêtes HTTP et inspecter les réponses des serveurs [42].

5.5 Implémentation du Modèle

Notre étude se concentre sur le développement et la mise en œuvre d'un modèle de recommandation qui utilise les données des utilisateurs pour fournir des recommandations financières précises et personnalisées. Pour implémenter notre système, nous utilisons un modèle d'arbre de décision, dont le fonctionnement dépend des informations contextuelles de l'utilisateur (montant d'argent disponible, préférences personnelles).

Il y a quatre étapes principales dans le processus : **préparation des données, Model Architecture Design, Model Training, production de recommandations** et **Conversion en TensorFlow Lite**

1. Préparation des données

Les données sont chargées à partir d'un fichier CSV dans un DataFrame pandas. La colonne cible est séparée des colonnes de caractéristiques en supprimant la colonne 'target' du DataFrame pour créer X pour les caractéristiques d'entrée et en sélectionnant la colonne 'target' pour créer y pour la variable cible. Les données sont ensuite divisées en ensembles d'entraînement et de test à l'aide de `train_test_split`, avec 80% des données allouées à l'entraînement et 20% au test. Cette étape garantit que le modèle peut être entraîné sur un sous-ensemble des données et évalué sur un autre, ce qui aide à prévenir le surapprentissage et à évaluer la performance du modèle sur des données non vues. Comme le représente la figure ()

```
import pandas as pd
from sklearn.model_selection import train_test_split

df = pd.read_csv('user_data.csv')

X = df.drop(columns=['target'])
y = df['target']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 26 : code pour la préparation des données

2. Model Architecture Design

La conception de l'architecture du modèle implique la définition d'un modèle séquentiel utilisant TensorFlow et Keras. La forme d'entrée est définie en fonction du nombre de caractéristiques dans les données d'entraînement. Le modèle comprend une couche d'entrée, une couche dense avec 128 unités et une fonction d'activation ReLU, suivie d'une couche dense avec une unité et une fonction d'activation sigmoïde pour produire une sortie de probabilité binaire. Le modèle est compilé avec l'optimiseur Adam, la perte de type `binary_crossentropy`, et la métrique d'exactitude. Comme le représente cette figure().

```
import tensorflow as tf

input_shape = X_train.shape[1]
keras_model = tf.keras.Sequential([
    tf.keras.layers.InputLayer(input_shape=input_shape),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

keras_model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Figure 27 : code pour model architecture

3. Model Training

L'entraînement du modèle est effectué en appelant la méthode `fit` du modèle Keras. Les données d'entraînement (`X_train` et `y_train`) sont utilisées pour entraîner le modèle pendant 200 époques avec une taille de lot de 32. Un fractionnement de validation de 20% est utilisé pour évaluer la performance du modèle pendant l'entraînement. Cela permet de surveiller les performances du modèle et d'éviter le surapprentissage. Que la figure() l'illustre.

```
# Model Training
keras_model.fit(X_train, y_train, epochs=200, batch_size=32, validation_split=0.2)
```

Figure 28 : code pour l'entrainement

4. Recommendation Generation

La génération de recommandations implique l'utilisation du modèle entraîné pour faire des prédictions sur de nouvelles données utilisateur. Une fonction `make_recommendation` est définie pour prendre en entrée un modèle et des données utilisateur, puis renvoyer une prédiction. Les valeurs des caractéristiques utilisateur doivent être fournies dans `user_data` pour obtenir des recommandations personnalisées.

```
import numpy as np

def make_recommendation(model, user_data):
    prediction = model.predict(user_data)
    return prediction

user_data = np.array([[34000, 23000, 'voyage', 'voiture']])
recommendation = make_recommendation(keras_model, user_data)
print(f"Recommendation: {recommendation}")
```

Figure 29 : code pour recommandation personnalisées

5. Convertir a TensorFlow Lite

Pour rendre le modèle utilisable sur des dispositifs mobiles ou des environnements à ressources limitées, il est converti en format TensorFlow Lite. Un convertisseur TensorFlow Lite est utilisé pour convertir le modèle Keras en un modèle TensorFlow Lite. Le modèle converti est ensuite enregistré dans un fichier nommé `model.tflite`. comme la figure () le démontre.

```
# Convert to TensorFlow Lite
converter = tf.lite.TFLiteConverter.from_keras_model(keras_model)
tflite_model = converter.convert()

with open('model.tflite', 'wb') as f:
    f.write(tflite_model)

print("Model has been converted to TensorFlow Lite and saved as 'model.tflite'")
```

Figure 30 : code pour conversion en tensorflow lite

5.6 Interface utilisateur

L'interface utilisateur (UI) d'une application de gestion de budget joue un rôle crucial dans l'expérience globale de l'utilisateur. Une UI bien conçue peut rendre l'application facile à utiliser, attrayante et motivante pour les utilisateurs, les encourageant ainsi à s'engager activement dans la gestion de leurs finances. À l'inverse, une UI mal conçue peut frustrer les utilisateurs, les dissuader d'utiliser l'application et, en fin de compte, les empêcher d'atteindre leurs objectifs financiers [44].

- **Page de connexion** : L'utilisateur arrive sur cette page lorsqu'il ouvre l'application pour la première fois ou après s'être déconnecté. Il doit entrer son email et son mot de passe pour accéder à l'application.

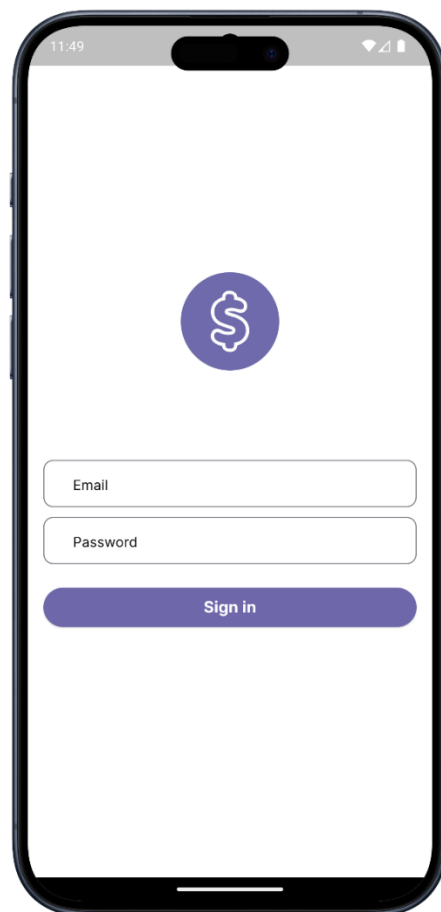


Figure 31 : page de connexion pour l'application Budget App

- **Page représentant le Menu** : Après s'être connecté avec succès l'utilisateur peut cliquer sur l'Icon menu qui affiche les autres fonctionnalités de l'application afin qu'il puisse sauter entre les pages.

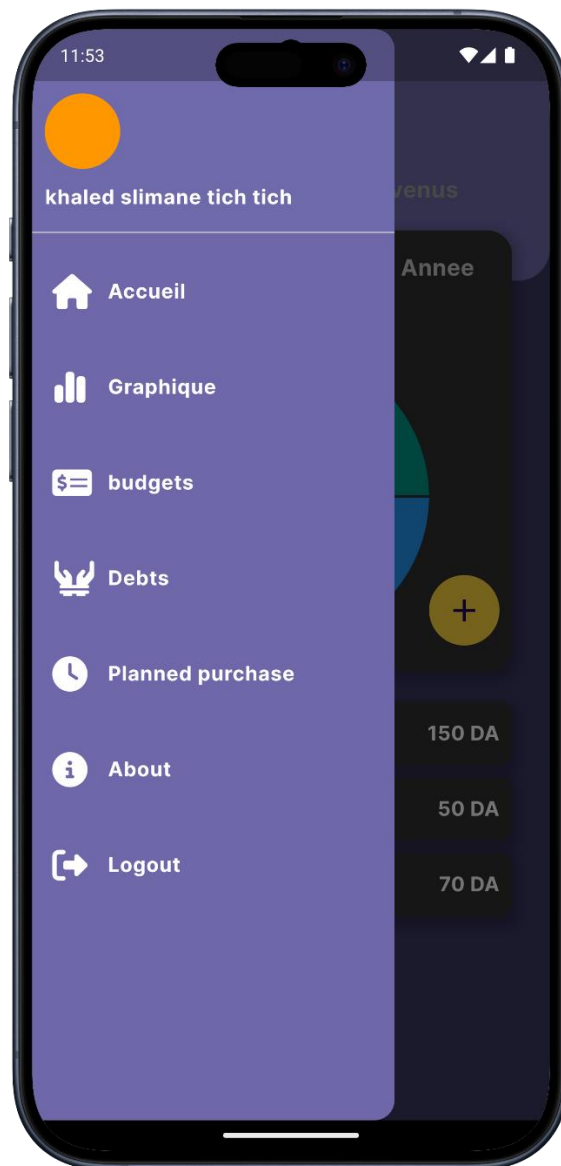


Figure 32 : Menu de l'application Budget App

- **Page d'Accueil :** Après s'être connecté avec succès, l'utilisateur est dirigé vers la page principale où il peut voir un aperçu de son état financier qui contient un graphe circulaire montrant la répartition des dépenses et des revenus, Liste des transactions récentes (dépenses et revenus) avec des filtres par jour, semaine, mois, année et un Bouton d'ajout de dépenses et de revenus.

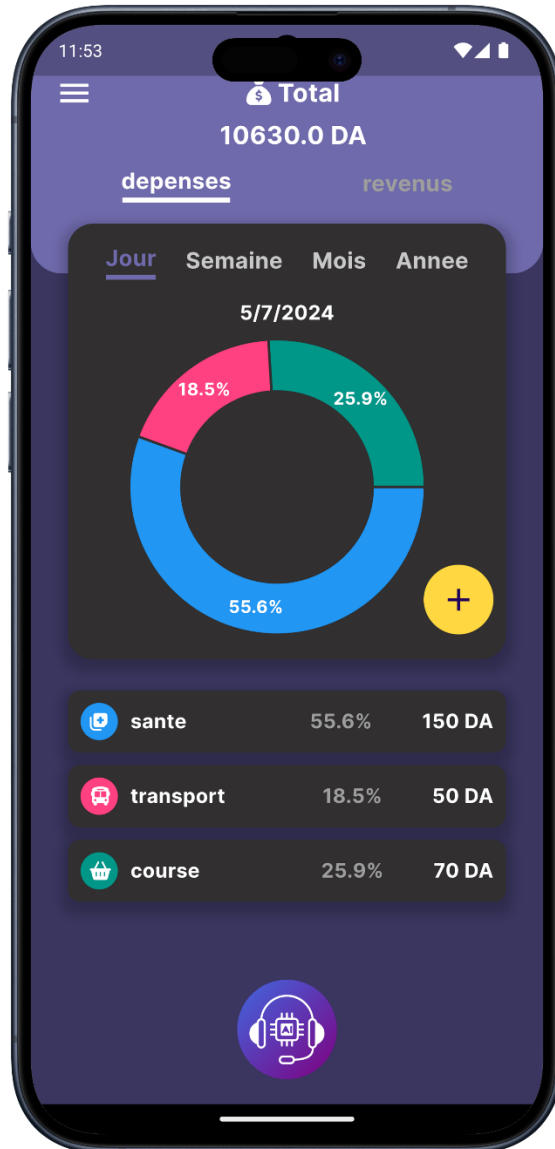


Figure 33 : page d'Accueil pour l'application Budget App

- **Page ajout de dépenses/revenus** : après avoir cliqué sur le bouton ajout dépense/revenus, l'utilisateur est dirigé vers la page où ajoute une dépense/revenu et met la valeur, la catégorie et la date.

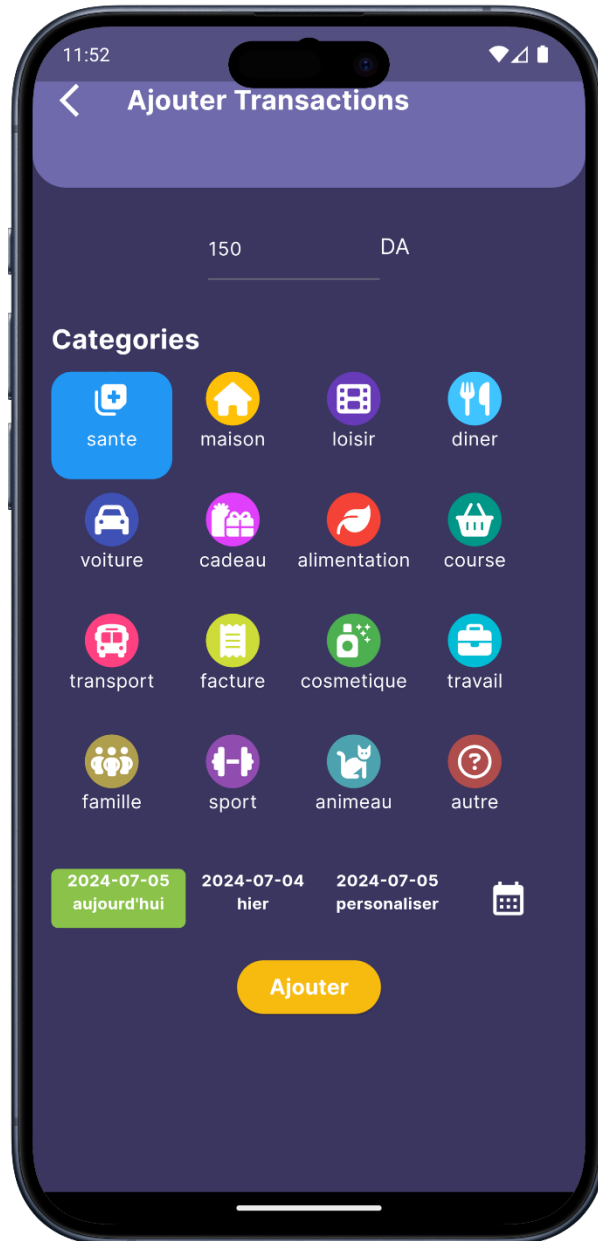


Figure 34 : page d'ajoute dépense/revenu pour l'application Budget App

- **Page des Budgets** : Permet à l'utilisateur de définir des budgets mensuels pour différentes catégories de dépenses. Qui contient Liste des budgets actuels avec des détails comme la catégorie, le montant fixe alloué, et la possibilité de les ajuster, Formulaire d'ajout de nouveaux budgets et Options pour modifier ou supprimer des budgets existants

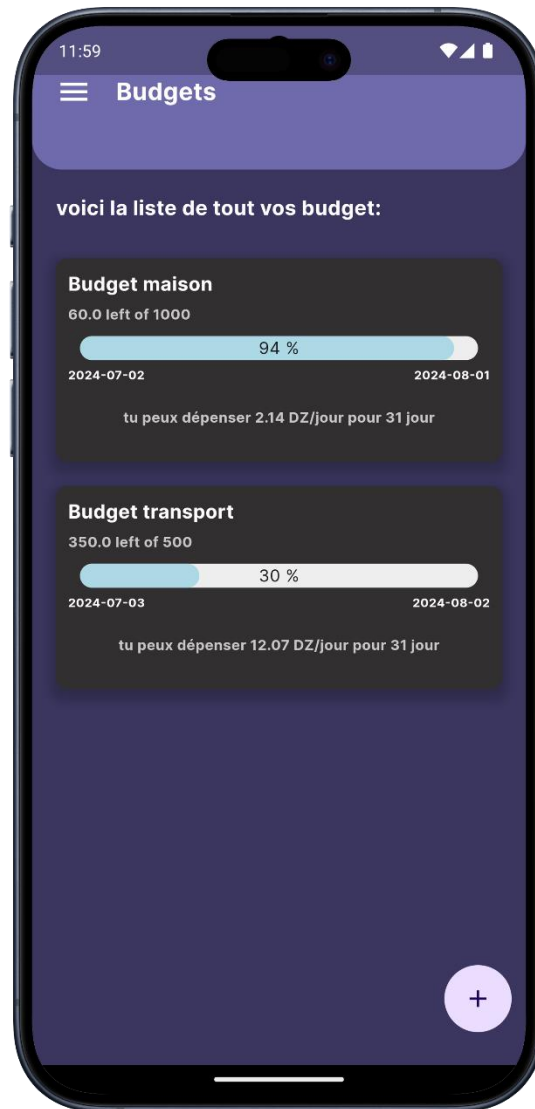


Figure 35 : page Budgets pour l'application Budget App

- Page des Dettes :** Permet à l'utilisateur de gérer les dettes ou les crédits qu'il doit rembourser et qui contient Liste des dettes ou crédits avec des détails comme le montant, la date d'échéance, et le type (dette ou crédit), Formulaire d'ajout de nouvelles dettes ou crédits et Options pour marquer une dette comme remboursée ou modifier les détails existants

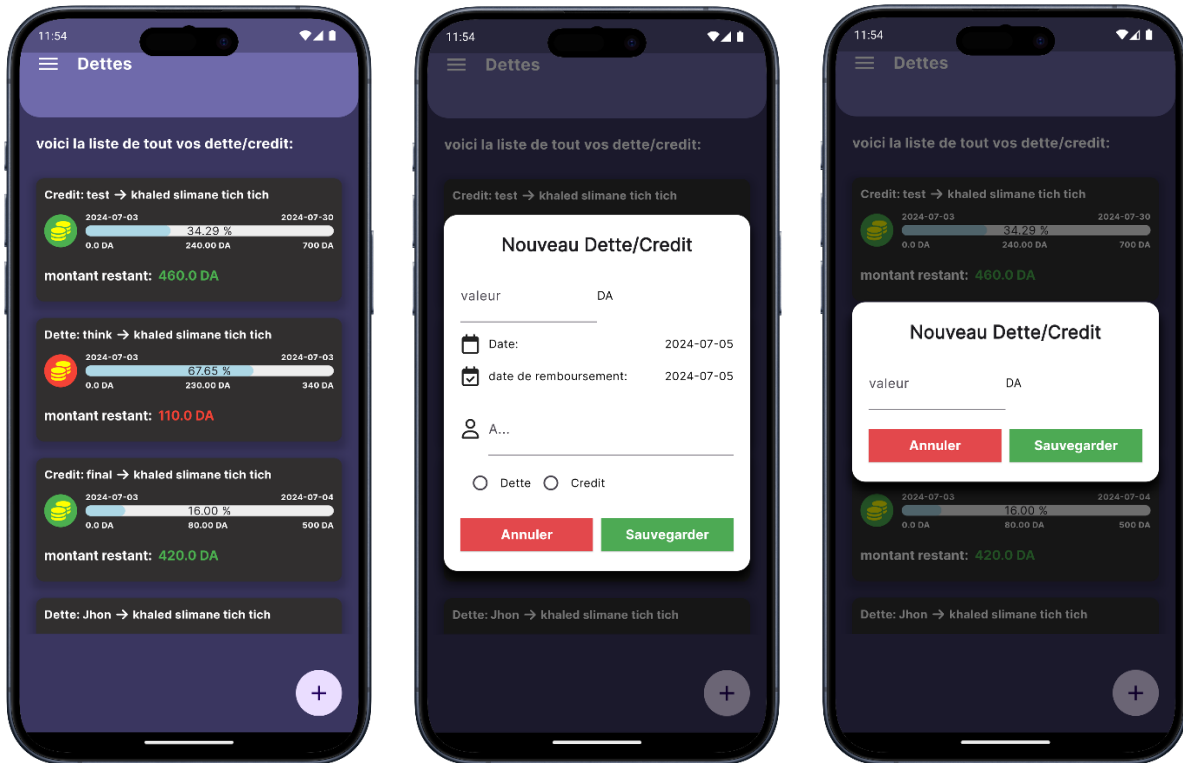


Figure 36 : page dettes/crédits pour l'application Budget App

- **Page graphiques** : l'utilisateur peut voir un aperçu de son état financier qui contient un graphe Barre montrant la répartition des dépenses et des revenus, avec des filtres par général, dépenses, revenus

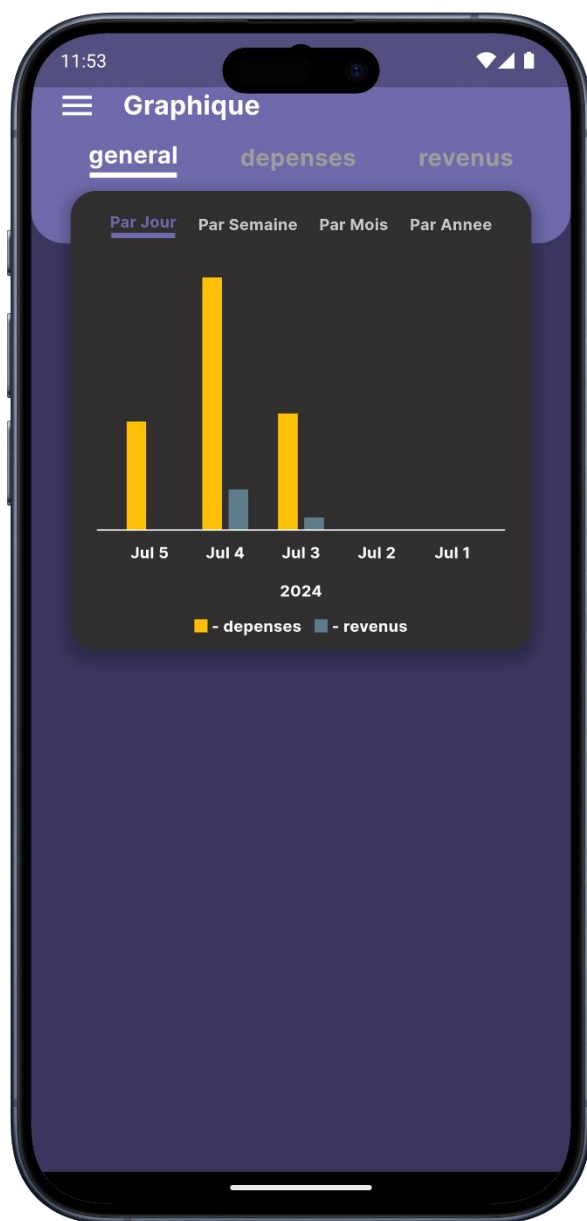


Figure 37 : page Graphique pour l'application Budget App

- Page Objectifs : l'utilisateur peut générer des objectifs automatiquement à l'aide de l'IA l'aidant ainsi à atteindre ses objectifs plus efficacement

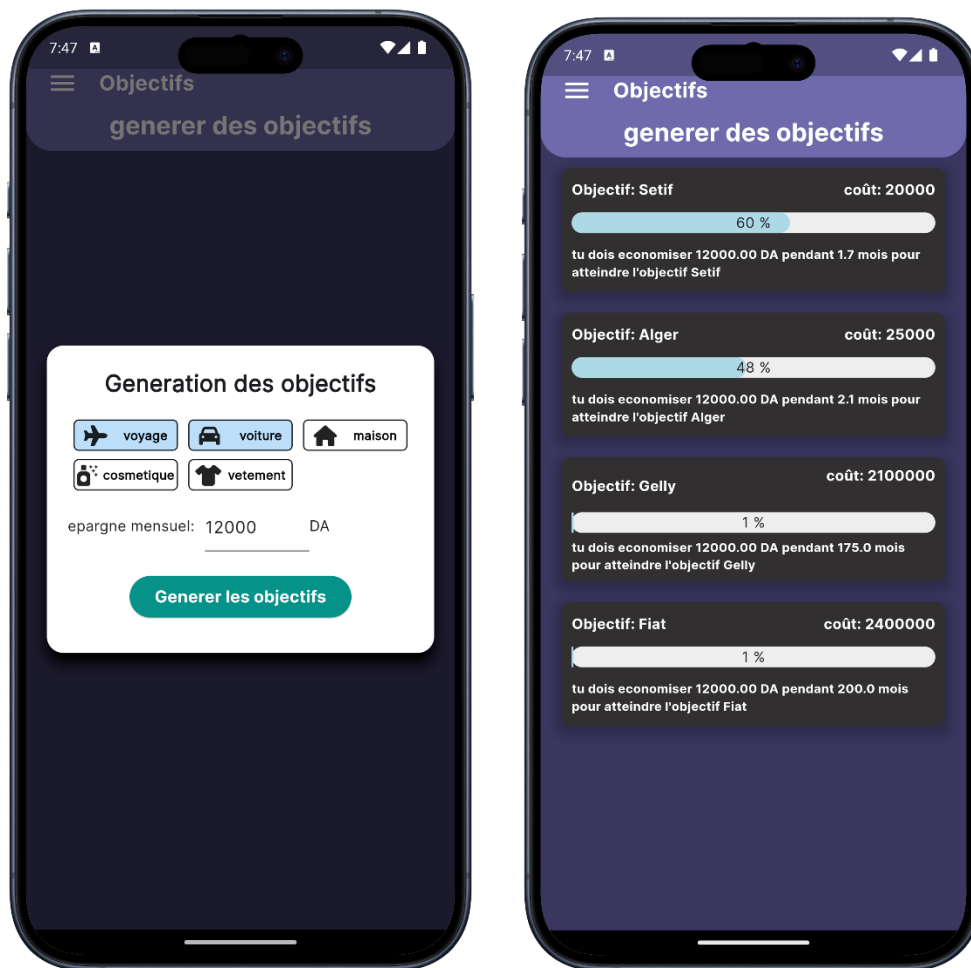


Figure 38 : génération des objectifs pour Budget App

5.7 Test

Comment avons-nous effectué nos tests ?

Trois évaluations distinctes ont été réalisées, une pour chaque catégorie niveau de revenu. Par la suite, une analyse a été réalisée pour déterminer si La suggestion s'aligne sur les préférences spécifiques de l'utilisateur en fonction des critères suivants :

- Si le niveau de revenu est faible affiche à l'utilisateur des recommandations des objectifs à faible coût et réduit le nombre de recommandation.
- Si le niveau de revenu est modéré il devrait avoir surtout des recommandations des objectifs avec un nombre moyen et un coût plus élevé.
- Si le niveau de revenu est élevé il devrait avoir la plupart des recommandations des objectifs.

Niveau de revenu	Nombre Total Utilisateurs	Recommandations Correctes	Recommandations Incorrectes	Précision (%)	Rappel (%)
Revenu Faible	100	73	27	73	73
Revenu Modéré	150	112	38	74	74
Revenu Élevé	85	66	19	77	77
Total/Moyenne	335	251	84	74	74

Tableau 2 : test de précision effectué sur des utilisateurs

5.8 Résultat statistique

Train Loss (Perte d'Entraînement) :

- Représente la performance du modèle sur les données d'entraînement.
- C'est une mesure utilisée pour évaluer à quel point le modèle s'adapte bien aux données d'entraînement.
- Pendant l'entraînement, le modèle reçoit des exemples du jeu de données d'entraînement et effectue des prédictions basées sur les modèles appris.
- La `train_loss` calcule la différence entre les prédictions du modèle et les valeurs cibles réelles pour chaque exemple d'entraînement.
- Une `train_loss` plus faible indique que le modèle apprend bien les motifs dans les données d'entraînement et fait des prédictions précises.

Val Loss (Perte de Validation) :

- Représente la performance du modèle sur un jeu de données de validation distinct.
- C'est crucial pour prévenir le surapprentissage, qui se produit lorsque le modèle mémorise trop bien les données d'entraînement mais se comporte mal sur des données non vues.
- Le jeu de données de validation n'est pas utilisé pour entraîner le modèle, mais plutôt pour évaluer sa capacité de généralisation - à quel point il performe sur de nouveaux exemples non vus.
- La `val_loss` est calculée de la même manière que la `train_loss`, mais sur les données de validation.
- Idéalement, tant la `train_loss` que la `val_loss` devraient diminuer pendant l'entraînement.
- Une grande différence entre la `train_loss` et la `val_loss` suggère que le modèle pourrait être en train de surapprendre.

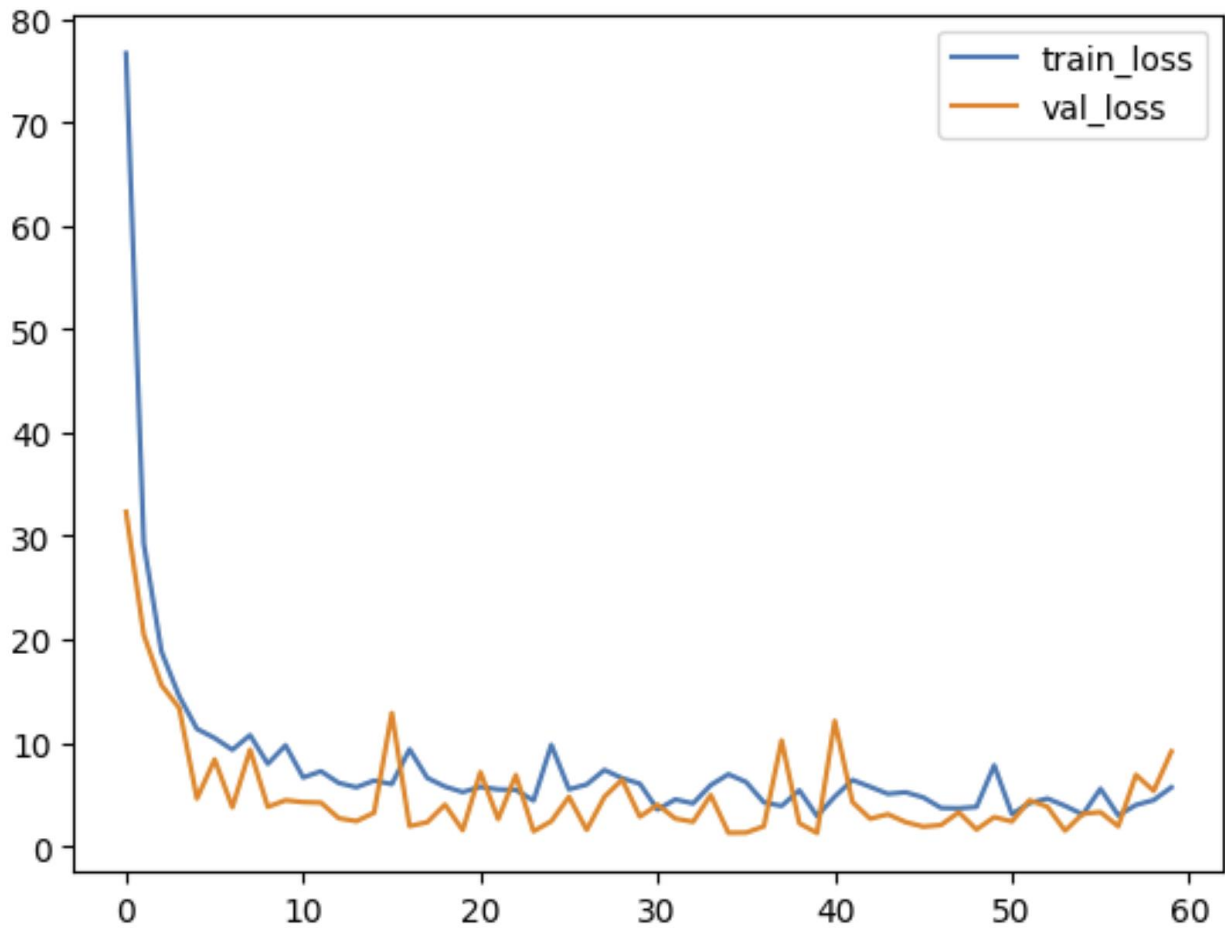


Figure 39 : statistique entre train_loss et val_loss pour le model de recommandation

5.9 Conclusion

Cette section a détaillé l'exécution et l'évaluation du système d'application de gestion budgétaire. Elle a examiné les outils et technologies utilisés dans le projet, ainsi qu'effectué une analyse approfondie des différentes phases de test. Les résultats soulignent l'efficacité et le potentiel du système. La section suivante explorera les développements potentiels de l'application de gestion budgétaire avant de conclure l'étude.

Conclusion Générale

L'application de gestion budgétaire représente une avancée significative dans la technologie financière, intégrant l'intelligence artificielle pour offrir aux utilisateurs des outils puissants pour la planification et la gestion financière efficaces. En exploitant les capacités de l'IA, l'application renforce sa capacité à fournir des recommandations et des insights personnalisés adaptés aux comportements, préférences et données financières individuelles des utilisateurs.

Tout au long de cette étude, nous avons exploré comment l'analyse alimentée par l'IA permet aux utilisateurs de prendre des décisions financières éclairées. La capacité de l'application à analyser de manière exhaustive les données financières des utilisateurs facilite des suggestions personnalisées pour la gestion de budget, la gestion des dépenses et l'établissement d'objectifs financiers. Cette approche personnalisée favorise non seulement un engagement plus profond des utilisateurs, mais maximise également la pertinence et l'efficacité des recommandations fournies.

Conçue avec une orientation centrée sur l'utilisateur, l'application propose des interfaces intuitives et des visualisations interactives qui simplifient le suivi des dépenses, la surveillance des revenus et la visualisation des tendances financières à travers des graphiques dynamiques. Cette conception conviviale contribue de manière significative à une expérience utilisateur fluide et enrichissante, améliorant l'accessibilité et l'utilisabilité auprès de diverses populations.

L'amélioration continue est un pilier de la stratégie de développement de l'application, avec des mises à jour continues et des améliorations itératives motivées par les commentaires des utilisateurs et les insights alimentés par l'IA. Cette approche agile garantit que l'application reste réactive aux besoins évolutifs des utilisateurs et aux dynamiques du marché, la positionnant comme un outil dynamique et adaptatif dans le domaine de la gestion financière.

À l'avenir, l'application est prête pour une croissance et une innovation significative. Les développements futurs pourraient voir une expansion des capacités de l'IA pour offrir des insights financiers prédictifs, une automatisation accrue de la catégorisation des dépenses et des ajustements budgétaires, ainsi qu'une intégration avec les technologies émergentes pour simplifier encore davantage les processus financiers.

En conclusion, l'application de gestion budgétaire représente bien plus qu'un simple outil de gestion financière - elle incarne une approche transformative de la planification financière personnalisée. En exploitant les technologies de l'IA, l'application permet aux utilisateurs de naviguer dans leur paysage financier avec confiance, établissant de nouveaux standards d'efficacité, d'efficience et d'autonomisation des utilisateurs dans la technologie financière.

Les Références

- [1] Nwamen, Fidèle. « Impact des technologies de l'information et de la communication sur la performance commerciale des entreprises », La Revue des Sciences de Gestion, vol. 218, no. 2, 2006, pp. 111-121.
- [2] Assadi, Djamchid, et Anaïs Cudi. « Le potentiel d'inclusion financière du “Mobile Banking”. Une étude exploratoire », Management & Avenir, vol. 46, no. 6, 2011, pp. 227-243.
- [3] Wataya, Roberto & Frauches, Patricia & Bergamo, Andressa. (2020). Finances personnelles dans la paume de votre main: un rapport d'expérience.
- [4] site : <https://sievo.com/fr/resources/spend-analysis-101>
- [5] <https://innowise.com/fr/blog/mobile-app-development-trends/>
- [6] <https://cba.ca/technology-innovation-banking?l=fr>
- [7] Alam, Azmir. (2023). What is Machine Learning?. 10.5281/zenodo.8231580.
- [8] Ayodele, Taiwo. (2010). Types of Machine Learning Algorithms. 10.5772/9385.
- [9] Rokach, Lior & Maimon, Oded. (2005). Decision Trees. 10.1007/0-387-25465-X_9.
- [10] Cutler, Adele & Cutler, David & Stevens, John. (2011). Random Forests. 10.1007/978-1-4419-9326-7_5.
- [11] Dayton, C.. (1992). Logistic Regression Analysis.

- [12] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.
- [13] Imandoust, S.B. & Bolandraftar, Mohammad. (2013). Application of K-nearest neighbor (KNN) approach for predicting economic events theoretical background. Int J Eng Res Appl. 3. 605-610.
- [14] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Eleventh annual conference of the international speech communication association, 2010.
- [15] Andrew Ng et al. Sparse autoencoder. CS294A Lecture notes, 72(2011):119, 2011.
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.
- [17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.
- [18] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15):2627–2636, 1998.
- [19] Rioult, François. (2011). Interprétation graphique de la courbe ROC. 301-302.
- [20] Matthieu Cornec. Validation croisée et modèles statistiques appliqués. Thèse dirigée par le Professeur Patrice Bertail, Université Paris X, Nanterre, 4 Juin 2009.

- [21] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET), pages 1–6. Ieee, 2017.
- [22] Naeem, Samreen & Ali, Aqib & Anam, Sania & Ahmed, Munawar. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. IJCDS Journal. 13. 911-921. 10.12785/ijcnds/130172.
- [23] OMG. (2017). Unified Modeling Language (UML), version 2.5.1. Object Management Group.
- [24] Ambler SW. The Elements of UMLTM 2.0 Style. Cambridge University Press; 2005.
- [25] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
- [26] <https://www.tensorflow.org/lite/guide>
- [27] Gossman, J. (2005). Introduction to Model/View/ViewModel pattern for building WPF apps. Microsoft Developer Network.
- [28] Dart Documentation. (n.d.). Dart Programming Language. Soustrait de <https://dart.dev/guides>
- [29] Python Documentation. soustrait de <https://www.python.org/doc/>
- [30] Flutter Documentation. Soustrait de <https://docs.flutter.dev/>
- [31] Supabase Documentation. (n.d.). Supabase - The Open Source Firebase Alternative. <https://supabase.com/docs>

- [32] Wit.ai Documentation. (n.d.). Wit.ai - Build applications that understand human language.
Soustrait de <https://wit.ai/docs>
- [33] TensorFlow Documentation. Soustrait de <https://www.tensorflow.org/tutorials?hl=fr>
- [34] Scikit-learn Documentation. (n.d.). Scikit-learn: Machine Learning in Python. <https://scikit-learn.org/>
- [35] Keras Documentation. (n.d.). Keras: The Python Deep Learning library.
<https://keras.io/guides/>
- [36] NumPy Documentation. (n.d.). NumPy. <https://numpy.org/doc/stable/>
- [37] Pandas Documentation. (n.d.). Pandas. <https://pandas.pydata.org/docs/>
- [38] Visual Studio Code Documentation. <https://code.visualstudio.com/docs>
- [39] Chacon, S., & Straub, B. (2014). Pro Git. Apress.
- [40] GitHub Documentation. (n.d.). GitHub. <https://github.com/topics/google-docs>
- [41] Figma Documentation. (n.d.). Figma. <https://www.figma.com/>
- [43] Postman Documentation. (n.d.). Postman. <https://www.postman.com/>
- [44] Dillon, Andrew. (2006). User Interface Design. 10.1002/0470018860.s00054.