

Sujets de Projets tutorés

Introduction

Un projet tutoré en analyse de données est une activité pédagogique au cours de laquelle les étudiants mettent en application leurs connaissances et compétences acquises en matière de collecte, traitement et analyse des données. Ce type de projet, encadré par un enseignant ou un tuteur, vise à développer à la fois des compétences techniques avancées en science des données et des compétences transversales comme la gestion de projet et la collaboration en équipe. Les projets tutorés en analyse de données offrent une opportunité unique de résoudre des problématiques réelles grâce à l'exploitation de données, tout en consolidant les concepts théoriques et les compétences pratiques. Ce projet est un tremplin pour les étudiants qui se destinent à une carrière dans le domaine de la science des données, de l'intelligence artificielle ou des systèmes décisionnels.

I. Contexte

Le projet tutoré s'adresse aux étudiants de troisième année en analyse de données et intelligence artificielle, qui se préparent à intégrer le marché du travail ou à effectuer un stage. Ce projet permet de consolider les compétences acquises durant les années de formation tout en confrontant les étudiants à des problématiques concrètes d'analyse et de traitement de données. Les étudiants devront traiter des jeux de données réels pour en extraire des informations utiles, construire des modèles prédictifs ou descriptifs, et présenter des résultats sous forme de tableaux de bord ou de rapports analytiques. L'objectif est de préparer les étudiants à manipuler des données dans un cadre professionnel et à maîtriser les outils, méthodes et bonnes pratiques liés à l'analyse de données.

II. Objectifs généraux

Les objectifs généraux du projet tutoré en analyse de données sont les suivants :

1. Appliquer les connaissances et compétences acquises en matière de collecte, prétraitement, analyse et visualisation des données.
2. Développer des compétences en modélisation de données, en intelligence artificielle, et en apprentissage automatique.
3. Se familiariser avec les outils et frameworks utilisés dans le domaine de l'analyse de données (Python, Pandas, Scikit-learn, etc.).
4. Acquérir des compétences transversales telles que la gestion de projet, la communication des résultats, et le travail en équipe.
5. Préparer les étudiants à leur future carrière dans le domaine de la science des données ou à des stages spécialisés.

III. Objectifs spécifiques

1. Collecter et préparer un jeu de données adapté au problème à résoudre.
2. Explorer et analyser les données pour en extraire des insights significatifs (analyse exploratoire).
3. Mettre en place des modèles d'analyse prédictive ou descriptive (modèles statistiques, machine learning).
4. Effectuer une validation rigoureuse des modèles et analyser leurs performances.
5. Visualiser les résultats et communiquer les analyses de manière compréhensible via des tableaux de bord et rapports.
6. Utiliser des outils de gestion de projet et de versionnage (Git, Trello) pour faciliter le travail en équipe et la collaboration.

IV. Résultats attendus

À la fin du projet tutoré, les résultats suivants sont attendus :

1. Une solution fonctionnelle d'analyse de données répondant à la problématique définie (modèles d'apprentissage automatique, rapports analytiques, etc.).
2. Un rapport complet incluant la description des données, les méthodes utilisées, les résultats obtenus et les conclusions tirées.
3. Une documentation détaillant l'architecture du projet, les méthodes employées, et les scripts utilisés pour la collecte, l'analyse et la modélisation des données.
4. Une présentation du projet mettant en évidence les conclusions tirées des analyses, les choix méthodologiques, et les recommandations basées sur les résultats.
5. Le développement des compétences techniques (manipulation de données, modélisation, visualisation) et transversales (travail en équipe, communication) chez les étudiants.

V. Description des projets

1. Chatbot de support client pour les services bancaires

Description

Développer un chatbot capable d'interagir avec les clients d'une banque pour répondre à des questions fréquentes, fournir des informations sur les comptes bancaires, les transactions, et même détecter des fraudes potentielles. Le chatbot utilisera des modèles de traitement du langage naturel (NLP) pour comprendre les requêtes et répondre intelligemment.

Objectif Général

Créer un chatbot de support client pour aider les utilisateurs à gérer leurs comptes bancaires et répondre à leurs questions.

Fonctionnalités Clés

- Compréhension et traitement des requêtes en langage naturel (NLP).
- Réponses automatiques aux questions fréquemment posées (FAQ).
- Gestion des requêtes liées aux transactions bancaires, solde de compte, et cartes bancaires.
- Détection de fraudes potentielles basée sur les interactions utilisateurs.
- Escalade des cas complexes vers un conseiller humain.

Technologies

- **Backend:** Flask (Python)
- **Frontend:** HTML/CSS avec un widget de chatbot.
- **NLP:** NLTK ou SpaCy pour la compréhension du langage.
- **Base de données:** PostgreSQL pour stocker les interactions utilisateurs.

Critères de Succès

- Capacité du chatbot à répondre correctement à au moins 80% des questions des utilisateurs.
- Rapidité et précision des réponses.
- Satisfaction des utilisateurs mesurée par des retours d'expérience intégrés dans l'application.

2. Chatbot pour le diagnostic médical préliminaire

Description

Développer un chatbot qui aide les utilisateurs à évaluer leurs symptômes et leur fournit des conseils médicaux préliminaires basés sur les données médicales. Le chatbot utilise l'apprentissage automatique et le NLP pour interpréter les symptômes décrits par l'utilisateur et propose une évaluation des risques.

Objectif Général

Créer un chatbot capable de fournir une évaluation médicale préliminaire basée sur les symptômes des utilisateurs.

Fonctionnalités Clés

- Saisie des symptômes en langage naturel.
- Analyse des symptômes à l'aide de techniques NLP.
- Prédiction des maladies possibles à l'aide de modèles de machine learning.
- Recommandations de soins et suggestions de consulter un médecin.
- Suivi des symptômes au fil du temps pour une meilleure évaluation.

Technologies

- **Backend:** Flask (Python)
- **Frontend:** HTML/CSS avec une interface utilisateur pour le chatbot.
- **NLP:** NLTK ou TensorFlow pour l'analyse des symptômes.
- **Machine Learning:** Scikit-learn pour l'analyse des données de santé.
- **Base de données:** MongoDB pour stocker les historiques des utilisateurs.

Critères de Succès

- Précision de l'évaluation préliminaire des symptômes.
- Utilisation efficace de l'apprentissage automatique pour fournir des diagnostics fiables.
- Interface conviviale permettant une saisie facile des symptômes.

3. Chatbot pour la gestion des ressources humaines de la fonction publique (RH)

Description

Développer un chatbot pour assister les employés dans la gestion de leurs tâches liées aux ressources humaines. Il aide à répondre aux questions sur les congés, la paie, les politiques internes, et à faire des demandes directement via le chatbot.

Objectif Général

Créer un chatbot RH qui répond aux questions des employés et les aide à gérer les tâches administratives.

Fonctionnalités Clés

- Réponses automatiques aux questions fréquentes concernant les congés, la paie, les politiques internes.
- Système de gestion des demandes de congés ou de remboursement.
- Notifications automatiques sur l'approbation des demandes et rappels d'échéances.
- Tableau de bord pour les RH permettant de suivre les interactions et d'améliorer les réponses du chatbot.

Technologies

- **Backend:** Flask (Python)
- **Frontend:** Bootstrap ou Vue.js pour une interface intuitive.
- **NLP:** NLTK pour la compréhension des requêtes en langage naturel.
- **Base de données:** PostgreSQL pour la gestion des demandes et des données RH.

Critères de Succès

- Réduction du nombre de requêtes RH traitées manuellement.
- Amélioration de l'efficacité de la gestion des tâches administratives pour les employés.
- Niveau de satisfaction des utilisateurs avec le système automatisé.

4. Application d'analyse de tendances sur Twitter

Description

Développer une application web qui extrait les tweets d'un ensemble de hashtags ou de mots-clés, analyse les tendances et génère des rapports d'analyse. Le système peut utiliser des techniques de traitement du langage naturel (NLP) pour classer les tweets en fonction des sentiments et identifier les sujets récurrents.

Objectif Général

Créer une plateforme qui scrape les tweets liés à des mots-clés spécifiques et les analyse pour détecter les tendances et sentiments dominants.

Fonctionnalités Clés

- Extraction des tweets en temps réel à partir de Twitter via l'API Twitter.
- Analyse des sentiments (positif, négatif, neutre) sur les tweets récupérés.

- Classement des tendances par fréquence de mots-clés ou de hashtags.
- Génération de graphiques pour visualiser les résultats de l'analyse.
- Exportation des résultats sous forme de rapports PDF ou CSV.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Tweepy pour l'interaction avec l'API Twitter.
- **NLP:** NLTK ou SpaCy pour l'analyse des sentiments.
- **Base de données:** MongoDB ou PostgreSQL pour stocker les tweets.
- **Visualisation:** Matplotlib ou Plotly pour générer des graphiques.

Critères de Succès

- Précision de l'analyse des sentiments sur les tweets.
- Capacité à extraire et analyser des volumes importants de tweets en temps réel.
- Interface utilisateur intuitive pour visualiser les tendances.

5. Application de surveillance des prix des produits en ligne

Description

Développer une application qui scrape les données de prix de plusieurs sites d'e-commerce (comme Jumia ou Amazon), puis analyse l'évolution des prix pour aider les utilisateurs à acheter au meilleur moment. L'application peut également envoyer des alertes lorsque le prix d'un produit atteint un seuil spécifique.

Objectif Général

Créer une application qui surveille et analyse les fluctuations des prix des produits sur différentes plateformes de vente en ligne.

Fonctionnalités Clés

- Scraping des pages de produits pour récupérer les prix actuels.
- Comparaison des prix d'un même produit sur plusieurs plateformes.
- Analyse des tendances de prix sur une période donnée.
- Système d'alertes pour notifier les utilisateurs lorsque le prix d'un produit baisse ou atteint un seuil défini.

- Tableau de bord pour visualiser les tendances et comparaisons de prix.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup ou Scrapy pour extraire les données des sites e-commerce.
- **Base de données:** SQLite ou PostgreSQL pour stocker les historiques de prix.
- **Notifications:** Intégration avec un service comme Twilio ou un système d'emailing pour envoyer des alertes.
- **Visualisation:** Plotly ou Matplotlib pour les graphiques.

Critères de Succès

- Précision et rapidité du scraping des prix.
- Capacité à gérer plusieurs plateformes simultanément.
- Pertinence des alertes de baisse de prix pour les utilisateurs.

6. Système de suivi des tendances de recrutement sur LinkedIn

Description

Développer une application web qui scrape les offres d'emploi et les tendances de recrutement sur LinkedIn, analyse les compétences demandées et fournit des statistiques sur les secteurs qui recrutent le plus. L'application peut être utilisée pour aider les étudiants et les professionnels à orienter leur carrière.

Objectif Général

Créer une application qui surveille les tendances de recrutement sur LinkedIn et fournit des statistiques sur les compétences les plus demandées par secteur.

Fonctionnalités Clés

- Scraping des offres d'emploi et des descriptions de poste sur LinkedIn.
- Analyse des compétences les plus fréquemment mentionnées.
- Visualisation des secteurs avec le plus d'opportunités.
- Suivi de l'évolution des tendances dans les exigences professionnelles.
- Génération de rapports personnalisés pour les utilisateurs.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Selenium pour interagir avec LinkedIn.
- **Base de données:** PostgreSQL pour stocker les données sur les offres d'emploi.
- **Data Analysis:** Pandas pour l'analyse des compétences et tendances.
- **Visualisation:** Dash ou Plotly pour les tableaux de bord interactifs.

Critères de Succès

- Pertinence des insights sur les compétences demandées.
- Capacité à suivre les tendances du marché du travail sur une période prolongée.
- Accessibilité de l'interface pour les utilisateurs souhaitant suivre les tendances de recrutement.

7. Chatbot pour la sensibilisation à la santé maternelle et infantile

Description

Ce chatbot a pour but de sensibiliser et d'informer les femmes enceintes et les jeunes mères sur les bonnes pratiques de santé maternelle et infantile. Il peut répondre à des questions sur la grossesse, les soins prénatals, la nutrition des enfants, et rappeler aux utilisateurs les dates de vaccination.

Objectif Général

Créer un chatbot de sensibilisation sur la santé maternelle et infantile pour répondre aux questions des femmes enceintes et jeunes mères et les informer sur les soins nécessaires.

Fonctionnalités Clés

- Réponses automatiques aux questions courantes sur la grossesse et les soins postnataux.
- Notifications et rappels des visites médicales et des vaccins pour les enfants.
- Système de gestion des consultations de santé à distance avec des conseillers.
- Suggestions personnalisées basées sur les informations saisies (ex. durée de la grossesse).

Technologies

- **Backend:** Flask (Python)

- **NLP:** NLTK ou SpaCy pour traiter les questions des utilisateurs.
- **Base de données:** MongoDB pour stocker les informations utilisateur et les consultations.
- **Notifications:** Twilio ou système de SMS pour envoyer des rappels de vaccination.

Critères de Succès

- Taux d'utilisation élevé parmi les femmes enceintes et jeunes mères.
- Amélioration de la sensibilisation et des suivis médicaux.
- Satisfaction des utilisateurs mesurée par les retours d'expérience.

8. Chatbot d'orientation des agriculteurs pour l'agriculture intelligente

Description

Un chatbot qui aide les agriculteurs à obtenir des conseils sur la gestion de leurs cultures et de leurs exploitations agricoles. Il pourrait donner des informations sur la meilleure période pour planter, les prévisions météorologiques locales, et les méthodes de lutte contre les parasites.

Objectif Général

Développer un chatbot qui assiste les agriculteurs dans la gestion de leurs cultures, en fournissant des conseils agricoles basés sur l'analyse des données climatiques et des pratiques locales.

Fonctionnalités Clés

- Réponses aux questions sur les périodes de plantation, la gestion des récoltes, et la lutte contre les maladies.
- Prévisions météorologiques personnalisées pour les régions rurales.
- Conseils basés sur les types de sols et cultures.
- Alertes en cas de risques météorologiques (ex. sécheresses, inondations).

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Utilisation d'API pour récupérer les données météorologiques.
- **NLP:** SpaCy pour traiter les requêtes des agriculteurs.

- **Base de données:** SQLite pour stocker les informations locales sur les cultures.

Critères de Succès

- Augmentation de la productivité des agriculteurs grâce aux conseils personnalisés.
- Taux d'adoption par les agriculteurs dans les zones rurales.
- Amélioration de la planification des récoltes en fonction des conditions météorologiques.

9. Chatbot pour la sensibilisation aux droits et lois

Description

Un chatbot conçu pour informer les citoyens sur leurs droits et sur les lois en vigueur au Burkina Faso. Il permet aux utilisateurs de poser des questions concernant leurs droits civils, les procédures légales, et fournit des informations sur les ressources juridiques disponibles.

Objectif Général

Créer un chatbot juridique capable d'informer les citoyens sur leurs droits, les procédures judiciaires et les lois nationales.

Fonctionnalités Clés

- Réponses aux questions courantes sur les droits civils, le droit du travail, et les lois foncières.
- Accès à des informations sur les démarches administratives et les services juridiques locaux.
- Suivi personnalisé des cas juridiques simples (ex. dépôt de plainte, obtention de documents).
- Notifications pour informer des évolutions législatives.

Technologies

- **Backend:** Flask (Python)
- **NLP:** NLTK ou SpaCy pour interpréter les questions juridiques.
- **Base de données:** PostgreSQL pour stocker les informations légales et les requêtes des utilisateurs.

Critères de Succès

- Niveau d'adoption par les citoyens cherchant des conseils juridiques.
- Précision des réponses aux questions juridiques.
- Taux de satisfaction mesuré par le nombre de cas résolus via le chatbot.

10. Chatbot pour l'orientation professionnelle et l'emploi

Description

Ce chatbot aide les jeunes diplômés et les chercheurs d'emploi à identifier des opportunités de carrière en fonction de leurs compétences et intérêts. Il peut fournir des recommandations de formation et d'emploi, et alerter sur les nouvelles offres disponibles sur le marché du travail au Burkina Faso.

Cahier des charges

Objectif Général

Créer un chatbot d'orientation professionnelle qui aide les utilisateurs à découvrir des opportunités d'emploi et à se former en fonction de leurs compétences.

Fonctionnalités Clés

- Analyse des compétences de l'utilisateur pour recommander des formations ou des offres d'emploi.
- Système de recherche d'emploi avec possibilité de filtrer les résultats par région ou secteur.
- Notifications des offres d'emploi correspondant aux compétences.
- Recommandations personnalisées pour des cours de formation professionnelle ou des stages.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Utilisation d'API ou scraping de plateformes d'emploi locales.
- **NLP:** SpaCy pour comprendre les compétences et les besoins des utilisateurs.
- **Base de données:** PostgreSQL pour stocker les profils et les opportunités d'emploi.

Critères de Succès

- Augmentation du taux d'emploi parmi les utilisateurs.

- Niveau d'engagement mesuré par le nombre d'utilisateurs récurrents.
- Efficacité des recommandations basées sur les compétences.

11. Chatbot pour l'information touristique et culturelle

Description

Un chatbot qui fournit des informations sur les sites touristiques, les événements culturels, et les lieux d'intérêt au Burkina Faso. Il aide les touristes à découvrir les meilleurs endroits à visiter et les événements à ne pas manquer.

Objectif Général

Créer un chatbot d'information touristique qui guide les visiteurs à travers les sites et événements culturels du Burkina Faso.

Fonctionnalités Clés

- Réponses automatiques aux questions sur les lieux touristiques et les événements culturels.
- Suggestions d'itinéraires de visite basées sur les préférences de l'utilisateur.
- Informations sur les services locaux (restaurants, hôtels, transport).
- Notifications sur les événements à venir (festivals, expositions).

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour comprendre les requêtes des utilisateurs.
- **Base de données:** PostgreSQL pour stocker les informations touristiques et culturelles.

Critères de Succès

- Taux d'adoption par les touristes et visiteurs locaux.
- Satisfaction des utilisateurs dans la planification de leurs visites.
- Niveau d'interaction avec le chatbot pour découvrir de nouveaux lieux et événements.

12. Plateforme de suivi des prix des produits de base sur les marchés locaux

Description

Cette application scrappe les données des marchés locaux et supermarchés au Burkina Faso pour suivre les prix des produits de base (comme le riz, le mil, le maïs). Elle permet aux utilisateurs de connaître les variations de prix et d'identifier les meilleurs moments pour acheter ces produits.

Objectif Général

Créer une plateforme qui scrape et analyse les prix des produits de base sur les marchés locaux pour informer les consommateurs et les commerçants.

Fonctionnalités Clés

- Scraping des sites web des marchés, supermarchés, ou applications de commerce local.
- Analyse des variations de prix sur une période donnée (quotidienne, hebdomadaire, mensuelle).
- Comparaison des prix entre différents points de vente.
- Notifications pour alerter les utilisateurs des hausses ou baisses de prix.
- Tableau de bord interactif pour visualiser les tendances des prix.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup ou Scrapy pour extraire les prix des produits.
- **Base de données:** PostgreSQL pour stocker les informations de prix et les tendances.
- **Visualisation:** Matplotlib ou Plotly pour les graphiques interactifs.

Critères de Succès

- Précision et actualisation fréquente des données de prix.
- Adoption de la plateforme par les commerçants et les consommateurs.
- Impact sur la gestion des achats des consommateurs.

13. Système de suivi des prévisions météorologiques pour l'agriculture

Description

Développer une application qui scrappe les données météorologiques des sites locaux et internationaux pour fournir des prévisions météorologiques spécifiques aux agriculteurs burkinabés. Elle aide les agriculteurs à planifier les semailles, les récoltes, et les traitements phytosanitaires.

Objectif Général

Développer un système de suivi des prévisions météorologiques qui aide les agriculteurs à optimiser la gestion de leurs cultures.

Fonctionnalités Clés

- Scraping des données météorologiques des sites tels que AccuWeather ou Météo Burkina.
- Prédictions personnalisées basées sur la localisation géographique des agriculteurs.
- Alertes pour les événements météorologiques extrêmes (sécheresses, inondations).
- Analyse des tendances climatiques sur des périodes plus longues (semaines, mois).
- Notifications pour recommander les meilleures périodes de semis et récolte.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour extraire les données météorologiques.
- **Base de données:** PostgreSQL pour stocker les prévisions et tendances.
- **Notifications:** Système de messagerie (SMS ou notifications mobiles).

Critères de Succès

- Précision des prévisions pour les régions rurales.
- Taux d'adoption par les agriculteurs.
- Impact positif sur la productivité agricole grâce aux recommandations basées sur la météo.

14. Application de suivi des offres immobilières locales

Description

Créer une plateforme web qui scrape les annonces immobilières de différentes plateformes locales et offre un tableau de bord pour suivre les tendances du marché immobilier au Burkina Faso. L'application permet aux utilisateurs de filtrer les annonces selon le type de bien, la localisation, et le prix.

Objectif Général

Développer une application web permettant aux utilisateurs de suivre et comparer les offres immobilières locales en temps réel.

Fonctionnalités Clés

- Scraping des sites d'annonces immobilières locales (ex. Jumia House, Vivastreet).
- Filtrage des annonces par type de bien (appartement, maison, terrain) et localisation.
- Analyse des tendances de prix par région ou quartier.
- Notifications pour les nouvelles annonces correspondant aux critères de recherche.
- Génération de rapports sur les tendances du marché immobilier.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Scrapy pour extraire les informations d'annonces immobilières.
- **Base de données:** MongoDB pour stocker les annonces et leurs métadonnées.
- **Visualisation:** Plotly ou Dash pour les tableaux de bord et graphiques.

Critères de Succès

- Pertinence des annonces affichées pour les utilisateurs.
- Précision des tendances de prix et comparaison entre quartiers.
- Taux d'engagement des utilisateurs avec la plateforme.

15. Plateforme de suivi des subventions et aides gouvernementales

Description

Cette application scrape les sites du gouvernement burkinabé et des organisations internationales pour suivre les programmes de subventions et d'aides destinées aux secteurs

comme l'agriculture, l'éducation, et la santé. Elle permet aux citoyens de s'informer sur les opportunités disponibles.

Objectif Général

Créer une plateforme qui suit et informe les utilisateurs des subventions et aides disponibles au Burkina Faso.

Fonctionnalités Clés

- Scraping des sites web gouvernementaux et d'organisations comme l'ONU ou la Banque Mondiale.
- Notification des nouvelles subventions et aides disponibles selon les secteurs.
- Système de recherche des programmes d'aide par secteur ou besoin spécifique.
- Tableau de bord pour suivre les opportunités de financement et les dates limites de candidature.
- Génération de rapports personnalisés pour les utilisateurs.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup ou Scrapy pour extraire les informations des sites officiels.
- **Base de données:** PostgreSQL pour stocker les données des subventions.
- **Notifications:** Twilio ou un système d'emailing pour informer les utilisateurs des nouvelles opportunités.

Critères de Succès

- Taux d'adoption par les citoyens et les petites entreprises cherchant des subventions.
- Impact sur l'accès des utilisateurs aux financements et subventions.
- Facilité d'utilisation et clarté des informations fournies.

16. Système d'analyse et de suivi des projets d'infrastructures au Burkina Faso

Description

Développer une plateforme web qui scrape les informations sur les projets d'infrastructure en cours au Burkina Faso (routes, écoles, hôpitaux, etc.) depuis les sites gouvernementaux et d'organisations partenaires. L'application aide les citoyens à suivre l'évolution des projets.

Objectif Général

Créer un système d'analyse et de suivi des projets d'infrastructure, permettant aux citoyens de suivre leur évolution et leur impact.

Fonctionnalités Clés

- Scraping des informations de projets d'infrastructure sur les sites gouvernementaux.
- Visualisation des projets par région, type d'infrastructure, et état d'avancement.
- Notifications pour informer les utilisateurs des mises à jour importantes sur les projets.
- Analyse des délais et budgets associés aux projets d'infrastructure.
- Génération de rapports sur l'avancement des projets.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Scrapy pour récupérer les informations des projets d'infrastructure.
- **Base de données:** PostgreSQL pour stocker les projets et leur état d'avancement.
- **Visualisation:** Plotly ou Matplotlib pour afficher les statistiques et graphiques.

Critères de Succès

- Transparence accrue sur les projets d'infrastructure.
- Engagement citoyen grâce à la visibilité des projets en cours.
- Amélioration de la communication entre le gouvernement et les citoyens.

17. Plateforme de suivi des politiques publiques et des débats parlementaires

Description

Créer une plateforme qui scrape les sites gouvernementaux et médiatiques pour suivre les politiques publiques et les débats parlementaires. Les citoyens peuvent ainsi suivre les décisions politiques et comprendre les implications des nouvelles lois et politiques.

Objectif Général

Développer une plateforme qui suit et analyse les politiques publiques et débats parlementaires au Burkina Faso.

Fonctionnalités Clés

- Scraping des débats parlementaires et des politiques publiques sur les sites gouvernementaux et médiatiques.
- Classement des débats par thème (éducation, économie, santé, etc.).
- Notifications pour informer les utilisateurs des débats à venir ou des décisions prises.
- Analyse des décisions politiques et leur impact potentiel sur les citoyens.
- Génération de rapports sur les principales lois et réformes en cours.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup ou Scrapy pour extraire les informations des sites officiels et des médias.
- **Base de données:** PostgreSQL pour stocker les débats et politiques publiques.
- **Visualisation:** Dash ou Plotly pour afficher les résultats sous forme de graphiques.

Critères de Succès

- Amélioration de la participation citoyenne dans les décisions politiques.
- Transparence accrue sur les débats parlementaires et décisions politiques.
- Taux d'engagement des citoyens avec les informations fournies.

18. Plateforme de gestion des opportunités de bourses d'études et de stages

Description

Une application qui scrappe les plateformes éducatives et les sites d'institutions pour suivre les offres de bourses d'études et de stages au Burkina Faso et à l'international. Elle aide les étudiants à trouver et postuler facilement aux opportunités disponibles.

Objectif Général

Créer une plateforme qui permet aux étudiants de suivre et de postuler aux offres de bourses d'études et de stages.

Fonctionnalités Clés

- Scraping des sites éducatifs locaux et internationaux pour extraire les offres de bourses et stages.
- Classement des opportunités par domaine d'études, critères de sélection et dates limites.
- Notifications des nouvelles opportunités correspondant aux profils des utilisateurs.
- Aide à la préparation des dossiers de candidature (documents requis, conseils, etc.).
- Suivi des candidatures et alertes sur les résultats.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour extraire les informations des sites éducatifs.
- **Base de données:** PostgreSQL pour gérer les offres de bourses et stages.
- **Notifications:** Intégration avec des services de messagerie pour informer les utilisateurs.

Critères de Succès

- Augmentation du nombre d'étudiants utilisant la plateforme pour postuler à des bourses.
- Facilitation de l'accès aux opportunités d'éducation et de stages.
- Amélioration des taux de succès des candidatures grâce aux conseils fournis.

19. Chatbot pour la promotion des langues locales au Burkina Faso

Description

Ce chatbot interactif vise à promouvoir et enseigner les langues locales du Burkina Faso (comme le mooré, dioula, fulfuldé). Il propose des leçons de vocabulaire, de grammaire et des quiz interactifs pour les utilisateurs.

Objectif Général

Créer un chatbot qui aide les utilisateurs à apprendre les langues locales grâce à des leçons interactives et des quiz.

Fonctionnalités Clés

- Leçons interactives de vocabulaire et grammaire pour les langues locales.
- Quiz pour tester les connaissances des utilisateurs et suivre leurs progrès.
- Notifications pour encourager une pratique régulière de la langue.
- Intégration de fichiers audio pour la prononciation correcte des mots.

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour l'analyse des réponses des utilisateurs.
- **Base de données:** PostgreSQL pour stocker les leçons et les scores des utilisateurs.

Critères de Succès

- Nombre d'utilisateurs réguliers apprenant les langues locales.
- Amélioration de la maîtrise des langues locales par les utilisateurs.
- Niveau d'engagement des utilisateurs avec les leçons et les quiz.

20. Système de détection de plagiat pour les universités

Description

Cette application web utilise des techniques de machine learning pour détecter le plagiat dans les travaux académiques des étudiants au Burkina Faso. Elle compare les documents soumis avec une base de données de travaux antérieurs, ainsi que des documents en ligne, pour identifier les similarités.

Objectif Général

Créer un système capable de détecter le plagiat dans les travaux académiques en utilisant des techniques d'analyse de texte.

Fonctionnalités Clés

- Comparaison des documents soumis avec une base de données interne et des sources en ligne.
- Analyse de similarité basée sur des techniques d'apprentissage automatique.
- Rapports détaillés sur les sections plagiées avec des liens vers les sources.
- Notifications automatiques pour alerter les enseignants en cas de plagiat détecté.

Technologies

- **Backend:** Flask (Python)
- **Machine Learning:** TensorFlow ou Scikit-learn pour la détection des similarités.
- **NLP:** NLTK pour l'analyse des documents textuels.
- **Base de données:** PostgreSQL pour stocker les travaux soumis.

Critères de Succès

- Réduction du taux de plagiat dans les universités.
- Efficacité dans la détection des similarités textuelles.
- Adoption par les enseignants pour vérifier les travaux des étudiants.

21. Application de suivi des coupures d'électricité et de gestion énergétique

Description

Développer une application qui scrape les données des coupures d'électricité et de gestion des ressources énergétiques au Burkina Faso. Elle permet aux utilisateurs de suivre les coupures, de recevoir des notifications et de planifier l'utilisation de générateurs ou autres sources d'énergie alternatives.

Objectif Général

Créer un système qui suit les coupures d'électricité et fournit des informations sur la gestion énergétique pour aider les ménages et entreprises à mieux planifier leur consommation.

Fonctionnalités Clés

- Scraping des informations sur les coupures d'électricité à partir des sites officiels (SONABEL, etc.).
- Notifications des coupures imminentes par région.
- Suivi de la durée des coupures et impact sur la consommation énergétique.
- Recommandations pour optimiser l'utilisation d'énergie alternative (solaire, générateurs).

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour récupérer les informations des sites d'énergie.

- **Base de données:** PostgreSQL pour stocker les coupures et l'état de la consommation énergétique.
- **Notifications:** Système de messagerie pour informer les utilisateurs.

Critères de Succès

- Impact sur la planification de l'utilisation de l'électricité.
- Amélioration de la gestion des ressources énergétiques dans les foyers et entreprises.
- Taux d'adoption de l'application dans les zones à forte fréquence de coupures.

22. Chatbot pour l'orientation des jeunes entrepreneurs

Description

Ce chatbot aide les jeunes entrepreneurs à obtenir des conseils sur la création d'entreprises, la gestion des finances, et l'accès aux subventions ou financements disponibles. Il peut fournir des ressources et des recommandations adaptées aux secteurs locaux en plein développement.

Objectif Général

Développer un chatbot qui assiste les jeunes entrepreneurs dans la création et la gestion de leurs entreprises en leur fournissant des conseils adaptés.

Fonctionnalités Clés

- Réponses aux questions fréquentes sur les démarches administratives, les finances, et les subventions.
- Recommandations personnalisées pour les entrepreneurs selon leurs secteurs d'activité.
- Notifications sur les opportunités de financement et les événements pour les startups.
- Suivi des progrès des entrepreneurs pour ajuster les conseils et les recommandations.

Technologies

- **Backend:** Flask (Python)

- **NLP:** SpaCy pour traiter les questions et fournir des conseils.
- **Base de données:** PostgreSQL pour stocker les informations sur les utilisateurs et leurs besoins.

Critères de Succès

- Aide à la création de nouvelles startups.
- Satisfaction des utilisateurs quant à la pertinence des conseils fournis.
- Impact sur le taux de succès des entrepreneurs locaux.

23. Système d'analyse des risques climatiques pour l'agriculture

Description

Développer une plateforme qui scrape les données météorologiques et environnementales pour analyser les risques climatiques (sécheresses, inondations) dans les zones agricoles du Burkina Faso. Elle aide les agriculteurs à anticiper les événements climatiques et à prendre des mesures pour protéger leurs cultures.

Objectif Général

Créer un système d'analyse des risques climatiques pour aider les agriculteurs à mieux se préparer aux événements météorologiques extrêmes.

Fonctionnalités Clés

- Scraping des données météorologiques et des prévisions à partir de sources fiables.
- Analyse des risques de sécheresse, inondation ou autres événements climatiques.
- Notifications pour alerter les agriculteurs des risques imminents.
- Recommandations sur les actions à entreprendre pour minimiser les pertes agricoles.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Scrapy ou BeautifulSoup pour extraire les données climatiques.
- **Machine Learning:** Scikit-learn pour l'analyse prédictive des risques.
- **Base de données:** PostgreSQL pour stocker les données climatiques et les analyses.

Critères de Succès

- Réduction des pertes agricoles liées aux risques climatiques.
- Amélioration de la productivité grâce à la gestion proactive des risques.

- Taux d'adoption de l'application parmi les agriculteurs.

24. Plateforme de suivi des tendances de l'emploi dans les secteurs émergents

Description

Cette plateforme scrappe les sites d'emploi et d'annonces pour analyser les tendances dans les secteurs émergents comme le numérique, l'énergie, et l'agro-industrie au Burkina Faso. Elle aide les chercheurs d'emploi à découvrir les compétences les plus demandées et les meilleures opportunités.

Objectif Général

Développer une plateforme qui analyse les tendances de l'emploi dans les secteurs émergents pour orienter les jeunes vers les compétences les plus recherchées.

Fonctionnalités Clés

- Scraping des sites d'annonces d'emploi pour récupérer les offres dans les secteurs émergents.
- Analyse des compétences les plus demandées par secteur.
- Notifications pour informer les utilisateurs des offres d'emploi correspondant à leur profil.
- Rapports sur les tendances du marché de l'emploi pour aider les décideurs à ajuster les programmes de formation.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour extraire les données des sites d'emploi.
- **Base de données:** PostgreSQL pour stocker les offres d'emploi et les compétences requises.
- **Visualisation:** Dash ou Plotly pour afficher les tendances.

Critères de Succès

- Augmentation des opportunités d'emploi pour les jeunes grâce à une meilleure orientation.
- Amélioration des programmes de formation en fonction des tendances du marché.

- Impact sur le taux d'emploi dans les secteurs émergents.

25. Chatbot d'information sur la santé publique et la prévention des maladies

Description

Développer un chatbot qui informe les utilisateurs sur la santé publique et la prévention des maladies au Burkina Faso. Il peut fournir des conseils sur les maladies fréquentes, les programmes de vaccination, et les mesures d'hygiène.

Objectif Général

Créer un chatbot qui informe les citoyens sur les bonnes pratiques en matière de santé publique et les aide à prévenir les maladies.

Fonctionnalités Clés

- Réponses aux questions sur la santé publique et la prévention des maladies courantes.
- Informations sur les programmes de vaccination et les dates de campagne.
- Conseils sur les mesures d'hygiène pour prévenir les épidémies.
- Notifications pour informer des campagnes de santé publique.

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour comprendre les questions en langage naturel.
- **Base de données:** PostgreSQL pour stocker les informations de santé publique.
- **Notifications:** Système d'alerte par SMS ou email.

Critères de Succès

- Amélioration de la sensibilisation à la santé publique et à la prévention des maladies.
- Taux d'adoption du chatbot parmi les citoyens.
- Impact sur la réduction des maladies transmissibles grâce aux conseils fournis.

VI. Cahiers de charges

1. Chatbot de support client pour les services bancaires

Objectif Général

Développer un chatbot de support client pour les banques, capable de répondre aux questions courantes des clients et de fournir des informations sur les services bancaires.

Fonctionnalités Clés

- Réponses automatiques aux questions fréquentes (FAQ).
- Accès aux informations de compte (solde, dernières transactions).
- Assistance pour la gestion des cartes bancaires (blocage, remplacement).
- Détection de fraudes potentielles et signalement.
- Escalade des cas complexes vers un conseiller humain.

Technologies

- Backend: Flask (Python)
- NLP: NLTK ou SpaCy pour la compréhension des requêtes.
- Base de données: PostgreSQL pour stocker les interactions utilisateurs.

Critères de Succès

- Taux élevé de résolution automatique des demandes clients.
- Précision dans la détection des fraudes.
- Satisfaction des utilisateurs.

2. Chatbot pour le diagnostic médical préliminaire

Objectif Général

Fournir un chatbot capable de réaliser un premier diagnostic médical basé sur les symptômes des utilisateurs.

Fonctionnalités Clés

- Saisie des symptômes en langage naturel.
- Analyse des symptômes pour identifier les maladies potentielles.
- Suggestions de soins ou conseils médicaux personnalisés.
- Notifications pour rappeler les rendez-vous médicaux et les traitements.

Technologies

- Backend: Flask (Python)
- NLP: SpaCy pour l'analyse des symptômes.
- Base de données: MongoDB pour stocker les profils utilisateurs et les diagnostics.
- Machine Learning: TensorFlow pour la prédiction des maladies.

Critères de Succès

- Précision des diagnostics préliminaires.
- Taux de satisfaction des utilisateurs.
- Sécurité des données médicales.

3. Chatbot pour la gestion des ressources humaines de la fonction publique (RH)

Objectif Général

Créer un chatbot capable de répondre aux questions des employés de la fonction publique sur les sujets liés aux ressources humaines.

Fonctionnalités Clés

- Réponses automatiques aux questions concernant les congés, la paie, et les avantages sociaux.
- Système de gestion des demandes de congés ou de remboursement.
- Notifications pour les échéances RH.
- Interface de suivi pour les gestionnaires RH.

Technologies

- Backend: Flask (Python)
- NLP: SpaCy pour comprendre les requêtes RH.
- Base de données: PostgreSQL pour gérer les données RH.

Critères de Succès

- Réduction des requêtes RH manuelles.
- Précision des réponses aux employés.
- Satisfaction des utilisateurs.

4. Application d'analyse de tendances sur Twitter

Objectif Général

Scraper les données de Twitter pour analyser les tendances basées sur des hashtags et des mots-clés spécifiques.

Fonctionnalités Clés

- Extraction des tweets via l'API Twitter.
- Analyse des sentiments (positif, négatif, neutre).
- Classement des tendances par mot-clé ou hashtag.
- Visualisation des résultats sous forme de graphiques.

Technologies

- Backend: Flask (Python)
- Scraping: Tweepy pour l'API Twitter.
- NLP: NLTK pour l'analyse des sentiments.
- Base de données: MongoDB pour stocker les tweets.

Critères de Succès

- Pertinence des tendances identifiées.
- Taux de satisfaction des utilisateurs.
- Précision de l'analyse des sentiments.

5. Application de surveillance des prix des produits en ligne

Objectif Général

Développer une application qui scrape et compare les prix des produits sur différentes plateformes e-commerce.

Fonctionnalités Clés

- Scraping des prix sur des sites e-commerce locaux et internationaux.
- Comparaison des prix et suivi des variations.
- Notifications pour les baisses de prix ou offres spéciales.
- Tableau de bord pour visualiser les tendances de prix.

Technologies

- Backend: Flask (Python)
- Scraping: BeautifulSoup ou Scrapy pour récupérer les prix.
- Base de données: PostgreSQL pour stocker les prix.
- Visualisation: Plotly pour les graphiques.

Critères de Succès

- Précision dans le suivi des prix.
- Satisfaction des utilisateurs avec les alertes de baisse de prix.
- Adoption par les consommateurs.

6. Système de suivi des tendances de recrutement sur LinkedIn

Objectif Général

Développer une application qui scrape les offres d'emploi sur LinkedIn et analyse les tendances de recrutement.

Fonctionnalités Clés

- Scraping des offres d'emploi sur LinkedIn.
- Analyse des compétences et qualifications demandées.
- Visualisation des tendances par secteur d'activité.
- Rapports pour les utilisateurs à la recherche d'emploi.

Technologies

- Backend: Flask (Python)
- Scraping: Selenium pour interagir avec LinkedIn.
- Base de données: PostgreSQL pour stocker les offres d'emploi.
- Visualisation: Dash ou Plotly.

Critères de Succès

- Précision des tendances de recrutement.
- Pertinence des recommandations de compétences.
- Satisfaction des utilisateurs à la recherche d'emploi.

7. Chatbot pour la sensibilisation à la santé maternelle et infantile

Objectif Général

Développer un chatbot qui informe et sensibilise les femmes enceintes et jeunes mères sur la santé maternelle et infantile.

Fonctionnalités Clés

- Réponses automatiques sur les questions de grossesse, soins prénataux et postnataux.
- Notifications pour rappeler les visites médicales et les vaccins des enfants.
- Conseils sur la nutrition et les soins des nourrissons.
- Suivi personnalisé des étapes de la grossesse.

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour comprendre les requêtes.
- **Base de données:** MongoDB pour stocker les profils des utilisateurs et les rappels.

Critères de Succès

- Impact sur la sensibilisation à la santé maternelle.
- Taux d'adoption par les mères.
- Satisfaction des utilisateurs mesurée par le nombre de consultations.

8. Chatbot d'orientation des agriculteurs pour l'agriculture intelligente

Objectif Général

Créer un chatbot qui aide les agriculteurs à prendre des décisions éclairées sur la gestion de leurs cultures en fonction des conditions locales.

Fonctionnalités Clés

- Réponses aux questions sur les périodes de semis et de récolte.
- Suggestions de méthodes agricoles basées sur les types de sols et cultures.
- Notifications des prévisions météorologiques pour les alertes de risques.
- Conseils sur la gestion des engrais et la lutte contre les parasites.

Technologies

- **Backend:** Flask (Python)
- **NLP:** NLTK ou SpaCy pour interpréter les questions des agriculteurs.
- **Base de données:** PostgreSQL pour stocker les données des cultures et sols.

Critères de Succès

- Amélioration des rendements agricoles.
- Adoption par les agriculteurs dans les zones rurales.
- Satisfaction des utilisateurs mesurée par l'efficacité des conseils.

9. Chatbot pour la sensibilisation aux droits et lois

Objectif Général

Développer un chatbot juridique pour informer les citoyens de leurs droits et des lois en vigueur au Burkina Faso.

Fonctionnalités Clés

- Réponses automatiques aux questions courantes sur le droit civil, le droit du travail et les procédures légales.
- Informations sur les démarches administratives et juridiques.
- Notifications des évolutions législatives et des réformes.
- Accès aux contacts d'avocats et de services juridiques locaux.

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour analyser les questions légales.
- **Base de données:** PostgreSQL pour stocker les lois et informations légales.

Critères de Succès

- Pertinence et précision des informations juridiques.
- Taux d'adoption par les citoyens.
- Impact sur l'accès aux services juridiques.

10. Chatbot pour l'orientation professionnelle et l'emploi

Objectif Général

Créer un chatbot pour aider les utilisateurs à trouver des opportunités d'emploi et de formation en fonction de leurs compétences.

Fonctionnalités Clés

- Analyse des compétences et recommandations d'emploi.
- Système de recherche d'emploi par région et secteur d'activité.
- Notifications pour informer des offres correspondant aux compétences des utilisateurs.
- Suggestions de formations pour améliorer les chances d'employabilité.

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour comprendre les compétences et les besoins des utilisateurs.
- **Base de données:** PostgreSQL pour stocker les offres d'emploi et les profils utilisateurs.

Critères de Succès

- Amélioration de l'accès aux opportunités d'emploi.
- Pertinence des recommandations de formation.
- Taux d'adoption par les utilisateurs en recherche d'emploi.

11. Chatbot pour l'information touristique et culturelle

Objectif Général

Développer un chatbot pour informer les utilisateurs sur les sites touristiques, événements culturels et lieux d'intérêt au Burkina Faso.

Fonctionnalités Clés

- Réponses automatiques sur les lieux touristiques, événements et festivals locaux.
- Suggestions d'itinéraires basées sur les préférences des utilisateurs.
- Notifications pour informer des événements à venir.
- Informations sur les services locaux (restaurants, hôtels, transport).

Technologies

- **Backend:** Flask (Python)
- **NLP:** NLTK ou SpaCy pour comprendre les requêtes des utilisateurs.
- **Base de données:** PostgreSQL pour stocker les informations touristiques et événements.

Critères de Succès

- Augmentation du nombre de touristes visitant les sites locaux.
- Impact sur la promotion de la culture locale.
- Taux d'adoption par les touristes et visiteurs locaux.

12. Plateforme de suivi des prix des produits de base sur les marchés locaux

Objectif Général

Créer une plateforme pour surveiller et analyser les prix des produits de base sur les marchés locaux au Burkina Faso.

Fonctionnalités Clés

- Scraping des sites des marchés locaux pour récupérer les prix des produits de base.
- Suivi des variations de prix sur une période donnée.
- Comparaison des prix entre différents marchés.
- Notifications pour alerter les utilisateurs des hausses ou baisses de prix.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Scrapy pour extraire les prix des produits.
- **Base de données:** PostgreSQL pour stocker les informations sur les prix.
- **Visualisation:** Plotly pour les graphiques.

Critères de Succès

- Précision des informations de prix.
- Impact sur la gestion des achats des consommateurs.
- Taux d'adoption par les commerçants et consommateurs.

13. Système de suivi des prévisions météorologiques pour l'agriculture

Objectif Général

Créer un système de suivi des prévisions météorologiques pour aider les agriculteurs à mieux planifier leurs activités agricoles.

Fonctionnalités Clés

- Scraping des données météorologiques des sources locales et internationales.
- Prévisions spécifiques aux régions agricoles du Burkina Faso.
- Notifications des alertes météorologiques (sécheresses, inondations).
- Suggestions pour optimiser la production en fonction des prévisions.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup ou Scrapy pour extraire les données météo.
- **Base de données:** PostgreSQL pour stocker les prévisions météorologiques.
- **Visualisation:** Dash ou Plotly pour afficher les prévisions et tendances.

Critères de Succès

- Précision des prévisions météorologiques.
- Impact sur la productivité agricole.
- Adoption par les agriculteurs.

14. Application de suivi des offres immobilières locales

Objectif Général

Développer une plateforme qui scrape les annonces immobilières et aide les utilisateurs à suivre et comparer les offres immobilières dans différentes régions du Burkina Faso.

Fonctionnalités Clés

- Scraping des plateformes d'annonces immobilières locales (Jumia House, Vivastreet).
- Filtrage des offres par type de bien (appartement, maison, terrain), prix, et localisation.
- Comparaison des prix de l'immobilier entre différentes régions.
- Notifications pour les nouvelles annonces correspondant aux critères de recherche.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Scrapy pour récupérer les annonces immobilières.
- **Base de données:** PostgreSQL pour stocker les informations des annonces.
- **Visualisation:** Plotly pour les graphiques de tendances de prix.

Critères de Succès

- Précision des annonces immobilières.
- Impact sur la transparence du marché immobilier.
- Taux d'adoption par les utilisateurs à la recherche de biens.

15. Plateforme de suivi des subventions et aides gouvernementales

Objectif Général

Créer une plateforme qui suit et informe les utilisateurs des subventions et aides disponibles dans divers secteurs (agriculture, éducation, santé) au Burkina Faso.

Fonctionnalités Clés

- Scraping des sites gouvernementaux et d'organisations internationales pour extraire les programmes de subventions.
- Système de recherche des subventions disponibles par secteur ou région.
- Notifications pour les nouvelles subventions correspondant aux critères des utilisateurs.
- Génération de rapports sur les opportunités de financement et dates limites.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour extraire les informations des sites officiels.
- **Base de données:** PostgreSQL pour stocker les informations sur les subventions et aides.
- **Visualisation:** Plotly pour afficher les opportunités disponibles.

Critères de Succès

- Accès accru aux subventions par les citoyens et les entreprises.
- Impact sur le développement communautaire grâce aux aides.
- Adoption par les utilisateurs cherchant des financements.

16. Système d'analyse et de suivi des projets d'infrastructures au Burkina Faso

Objectif Général

Créer une plateforme de suivi des projets d'infrastructure (routes, écoles, hôpitaux) au Burkina Faso pour informer les citoyens de l'avancement des travaux.

Fonctionnalités Clés

- Scraping des informations sur les projets d'infrastructure depuis les sites gouvernementaux et de partenaires internationaux.
- Suivi de l'état d'avancement des projets (délai, budget).
- Notifications pour alerter les citoyens des projets achevés ou retardés.
- Génération de rapports sur l'impact des projets par région.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** Scrapy pour extraire les informations sur les projets.
- **Base de données:** PostgreSQL pour stocker les informations des projets et leur progression.
- **Visualisation:** Dash pour visualiser les données des projets.

Critères de Succès

- Transparence accrue sur les projets d'infrastructure.
- Engagement citoyen renforcé.

- Impact sur la gestion des ressources par les autorités.

17. Plateforme de suivi des politiques publiques et des débats parlementaires

Objectif Général

Développer une plateforme qui scrape et suit les politiques publiques et débats parlementaires au Burkina Faso, afin d'informer les citoyens des décisions prises.

Fonctionnalités Clés

- Scraping des informations sur les débats parlementaires et politiques publiques.
- Classement des débats par thème (santé, économie, éducation).
- Notifications pour informer les utilisateurs des débats à venir.
- Visualisation des décisions politiques et leur impact potentiel.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup ou Scrapy pour extraire les informations des sites gouvernementaux.
- **Base de données:** PostgreSQL pour stocker les débats et politiques publiques.
- **Visualisation:** Plotly pour afficher les résultats des débats.

Critères de Succès

- Amélioration de la transparence sur les politiques publiques.
- Engagement citoyen dans les décisions politiques.

- Adoption par les utilisateurs intéressés par les affaires publiques.

18. Plateforme de gestion des opportunités de bourses d'études et de stages

Objectif Général

Créer une plateforme qui aide les étudiants à suivre et postuler aux bourses d'études et stages, en scrappant les opportunités disponibles localement et internationalement.

Fonctionnalités Clés

- Scraping des sites d'institutions éducatives et d'organisations internationales pour les offres de bourses et stages.
- Classement des opportunités par niveau d'études, domaine et date limite.
- Notifications pour les nouvelles offres correspondant au profil des utilisateurs.
- Assistance à la préparation des dossiers de candidature.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour extraire les informations sur les bourses et stages.
- **Base de données:** PostgreSQL pour stocker les offres et candidatures.
- **Notifications:** Système d'alerte pour informer les utilisateurs des nouvelles opportunités.

Critères de Succès

- Augmentation du nombre de candidatures réussies.

- Facilitation de l'accès aux opportunités éducatives.
- Adoption par les étudiants et jeunes professionnels.

19. Chatbot pour la promotion des langues locales au Burkina Faso

Objectif Général

Développer un chatbot qui enseigne les langues locales du Burkina Faso (mooré, dioula, fulfuldé) à travers des leçons interactives et des quiz.

Fonctionnalités Clés

- Leçons interactives de vocabulaire et de grammaire pour les langues locales.
- Quiz pour tester les connaissances des utilisateurs.
- Notifications pour rappeler les leçons quotidiennes.
- Intégration de fichiers audio pour apprendre la prononciation.

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour l'analyse des réponses des utilisateurs.
- **Base de données:** PostgreSQL pour stocker les leçons et les résultats des quiz.

Critères de Succès

- Augmentation du nombre de personnes apprenant les langues locales.
- Amélioration de la maîtrise des langues locales par les utilisateurs.

- Taux d'engagement élevé avec les leçons et quiz.

20. Système de détection de plagiat pour les universités

Objectif Général

Développer une application web qui détecte le plagiat dans les travaux académiques soumis par les étudiants, en comparant les documents avec une base de données interne et des sources en ligne.

Fonctionnalités Clés

- Système de soumission de travaux académiques.
- Analyse de similarité avec une base de données locale et des sources externes.
- Rapports détaillés sur les sections plagiées avec des références aux sources.
- Notifications pour alerter les enseignants en cas de plagiat détecté.

Technologies

- **Backend:** Flask (Python)
- **Machine Learning:** Scikit-learn ou TensorFlow pour l'analyse des similarités textuelles.
- **NLP:** NLTK pour l'analyse des documents.
- **Base de données:** PostgreSQL pour stocker les travaux et les résultats de l'analyse.

Critères de Succès

- Réduction du taux de plagiat dans les universités.

- Adoption par les enseignants pour vérifier les travaux soumis.
- Taux de satisfaction des enseignants quant à la précision des rapports.

21. Application de suivi des coupures d'électricité et de gestion énergétique

Objectif Général

Développer une application qui surveille les coupures d'électricité et aide les utilisateurs à planifier leur consommation d'énergie, en proposant des solutions alternatives.

Fonctionnalités Clés

- Scraping des données sur les coupures d'électricité des fournisseurs locaux.
- Notifications pour alerter les utilisateurs des coupures à venir.
- Suivi en temps réel des coupures par région.
- Recommandations pour la gestion des ressources énergétiques alternatives (générateurs, panneaux solaires).

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour récupérer les informations des sites de distribution d'électricité.
- **Base de données:** PostgreSQL pour stocker les informations sur les coupures.
- **Notifications:** Système d'alerte pour informer les utilisateurs.

Critères de Succès

- Amélioration de la gestion de l'énergie pendant les coupures.
- Adoption par les ménages et entreprises.
- Taux de satisfaction des utilisateurs.

22. Chatbot pour l'orientation des jeunes entrepreneurs

Objectif Général

Développer un chatbot pour conseiller les jeunes entrepreneurs dans la création et la gestion de leurs entreprises, en fournissant des recommandations adaptées à leur secteur d'activité.

Fonctionnalités Clés

- Réponses automatiques aux questions sur les démarches administratives, les financements, et la gestion d'entreprise.
- Suggestions de stratégies pour le lancement et la croissance d'une startup.
- Notifications pour informer des événements, financements, et opportunités d'accompagnement entrepreneurial.
- Suivi des progrès des entrepreneurs et ajustement des recommandations.

Technologies

- **Backend:** Flask (Python)
- **NLP:** SpaCy pour comprendre les questions des utilisateurs.
- **Base de données:** PostgreSQL pour stocker les profils des entrepreneurs et leurs besoins.

Critères de Succès

- Impact sur la création de startups.
- Satisfaction des utilisateurs grâce aux conseils adaptés.
- Taux d'adoption par les jeunes entrepreneurs.

23. Système d'analyse des risques climatiques pour l'agriculture

Objectif Général

Développer un système d'analyse qui prédit les risques climatiques (sécheresse, inondations) et fournit des recommandations aux agriculteurs pour protéger leurs cultures.

Fonctionnalités Clés

- Scraping des données météorologiques des sources locales et internationales.
- Modèles de prévision des événements climatiques extrêmes.
- Notifications pour alerter les agriculteurs des risques à venir.
- Recommandations sur les meilleures pratiques agricoles basées sur les prévisions.

Technologies

- **Backend:** Flask (Python)
- **Machine Learning:** Scikit-learn pour les modèles de prédiction climatique.
- **Scraping:** BeautifulSoup pour extraire les données météorologiques.
- **Base de données:** PostgreSQL pour stocker les données climatiques.

Critères de Succès

- Réduction des pertes agricoles dues aux événements climatiques.
- Adoption par les agriculteurs pour gérer les risques.
- Précision des prédictions et pertinence des recommandations.

24. Plateforme de suivi des tendances de l'emploi dans les secteurs émergents

Objectif Général

Développer une plateforme qui scrape les offres d'emploi dans les secteurs émergents au Burkina Faso (numérique, énergie renouvelable) pour aider les chercheurs d'emploi à s'orienter vers les compétences les plus demandées.

Fonctionnalités Clés

- Scraping des offres d'emploi dans les secteurs émergents (numérique, énergie).
- Analyse des compétences les plus recherchées par secteur.
- Rapports sur les tendances de recrutement par région et secteur.
- Notifications pour informer les utilisateurs des offres correspondant à leur profil.

Technologies

- **Backend:** Flask (Python)
- **Scraping:** BeautifulSoup pour extraire les offres d'emploi.
- **Base de données:** PostgreSQL pour stocker les offres d'emploi et compétences.
- **Visualisation:** Dash ou Plotly pour afficher les tendances.

Critères de Succès

- Augmentation de l'accès aux emplois dans les secteurs émergents.
- Impact sur la formation et l'orientation professionnelle.
- Adoption par les jeunes chercheurs d'emploi.

25. Chatbot d'information sur la santé publique et la prévention des maladies

Objectif Général

Créer un chatbot pour informer les citoyens sur les bonnes pratiques en matière de santé publique et les aider à prévenir les maladies courantes au Burkina Faso.

Fonctionnalités Clés

- Réponses automatiques aux questions sur la prévention des maladies (paludisme, choléra, etc.).
- Informations sur les campagnes de vaccination et programmes de santé publique.
- Notifications pour informer des campagnes de sensibilisation.
- Conseils personnalisés pour améliorer l'hygiène et la santé publique.

Technologies

- **Backend:** Flask (Python)
- **NLP:** NLTK ou SpaCy pour comprendre les questions sur la santé publique.
- **Base de données:** PostgreSQL pour stocker les informations sur les maladies et les campagnes de prévention.

Critères de Succès

- Amélioration de la sensibilisation à la santé publique.
- Réduction des maladies transmissibles grâce à la prévention.
- Taux d'adoption du chatbot par les citoyens.

VII. Canevas de rédaction du rapport du projet tutoré

1. Introduction

- **Contexte du projet**
 - Présentation générale du domaine (ex : data science, IA, etc.).
 - Problématique à résoudre avec l'analyse de données.
- **Objectifs du projet**
 - Identification des objectifs à atteindre à travers l'analyse de données.
 - Impact attendu de l'analyse sur la prise de décision ou les processus à améliorer.
- **Description de la plateforme ou des outils utilisés**
 - Présentation des outils ou plateformes de collecte et traitement des données.

2. Analyse des besoins

- **Description des données collectées**
 - Origine des données (source de données, scraping, bases externes).
 - Volume et type de données (structurées, non structurées, big data, etc.).
- **Analyse des utilisateurs cibles et de leurs besoins**
 - Identification des bénéficiaires de l'analyse de données (clients, entreprises, secteur public).
 - Besoins spécifiques des utilisateurs en termes d'insights ou de prédictions.
- **Étude de marché et concurrence**
 - Analyse des solutions existantes utilisant des données similaires.
 - Positionnement du projet par rapport aux approches existantes.

3. Préparation et Exploration des Données

- **Pré-traitement des données**
 - Nettoyage des données (suppression des valeurs manquantes, correction des anomalies, etc.).
 - Transformation des données (normalisation, agrégation, etc.).
- **Exploration des données**
 - Analyse exploratoire des données (distribution, corrélations, outliers, etc.).
 - Visualisation initiale des données (graphiques, tableaux).
- **Feature Engineering**
 - Création de nouvelles variables pertinentes pour l'analyse ou les modèles.
 - Sélection des features les plus importantes pour l'analyse.

4. Conception et Modélisation

- **Choix des techniques d'analyse et des modèles**
 - Justification des méthodes statistiques, algorithmes d'apprentissage machine ou techniques de deep learning choisies.
 - Présentation des alternatives et des choix finaux.
- **Description de l'architecture du système d'analyse**
 - Schéma des processus d'analyse de bout en bout (collecte, traitement, modélisation, visualisation).
 - Modélisation des données (diagramme des flux de données, etc.).
- **Modélisation de la base de données (si applicable)**
 - Schéma de la base de données utilisée pour stocker les résultats de l'analyse.

5. Implémentation des modèles d'analyse

- **Explication des étapes de développement**
 - Présentation des différentes étapes de la mise en place des modèles et du pipeline d'analyse.
- **Présentation des modèles et algorithmes développés**
 - Détails sur les modèles implémentés (algorithmes de machine learning, IA).
 - Explication des paramètres et hyperparamètres utilisés.
- **Description des modules et composants clés**

- Explication des modules ou scripts responsables des étapes majeures (pré-traitement, prédiction, visualisation).

6. Résultats de l'analyse

- **Résultats des modèles et interprétation**
 - Performance des modèles d'analyse (précision, rappel, F1-score, etc.).
 - Analyse des résultats obtenus par rapport aux objectifs fixés.
- **Visualisation des résultats**
 - Graphiques, tableaux, et autres moyens de présentation des résultats.
 - Comparaison des résultats des différents modèles.

7. Tests et Validation

- **Description des méthodes de validation**
 - Validation croisée, tests A/B, division en jeux de données d'entraînement et de test.
- **Résultats des tests et analyse des erreurs**
 - Analyse des erreurs et limites des modèles (erreurs de prédiction, biais, variance).
 - Propositions d'amélioration basées sur les résultats.

8. Performances et Optimisation

- **Évaluation des performances des modèles**
 - Temps de calcul, efficacité des algorithmes sur de grands ensembles de données.
- **Optimisation des modèles**
 - Ajustements des hyperparamètres pour améliorer les performances.
 - Techniques d'optimisation utilisées (pruning, grid search, etc.).

9. Sécurité et protection des données

- **Mesures de sécurité des données**
 - Explication des techniques de protection des données (cryptage, anonymisation).
 - Respect des normes de confidentialité (RGPD, protection des données sensibles).
- **Gestion des données personnelles**

- Politique de stockage et de traitement des données personnelles, si applicable.

10. Conclusion

- **Résumé des réalisations du projet**
 - Synthèse des résultats obtenus et des objectifs atteints.
- **Perspectives d'amélioration**
 - Pistes d'amélioration des modèles et techniques d'analyse utilisées.
- **Développement futur**
 - Opportunités d'évolution et d'intégration à plus grande échelle.

11. Annexes

- **Diagrammes et visualisations complémentaires**
 - Graphiques supplémentaires issus de l'analyse de données.
- **Extraits de code source**
 - Code clé utilisé pour le traitement des données et les modèles.
- **Documentation des algorithmes et modèles**
 - Documentation technique complémentaire (modèles statistiques, architecture, etc.).