

CATÉGORISATION AUTOMATIQUE DES TICKETS

INTRODUCTION & CONTEXTE DU PROJET



Aujourd'hui, les entreprises gèrent un très grand volume de demandes clients et internes via des plateformes de ticketing.

Cependant, une grande partie de ces tickets sont mal catégorisés manuellement, ce qui complique le tri, le traitement et la réaffectation.

Le temps de résolution augmente, les équipes perdent en efficacité, et l'expérience utilisateur se dégrade.

► Notre solution : développer un outil intelligent capable de catégoriser automatiquement les tickets selon leur contenu, en utilisant des techniques d'intelligence artificielle et de traitement du langage naturel (NLP).

OBJECTIFS FONCTIONNELS

1. Classification Automatique: Le système analyse le titre et la description d'un ticket pour lui attribuer automatiquement la catégorie la plus pertinente (ex: "Problème réseau", "Support messagerie").
- 2-Interface Utilisateur Intuitive: Une interface ergonomique permet de visualiser la catégorie attribuée, de la corriger manuellement si nécessaire, et d'avoir un aperçu rapide des tickets
- 3-Réduire les délais de traitement des tickets mal catégorisés : L'automatisation permet de réduire de le temps de tri manuel, en évitant les longues lectures et en affectant immédiatement le ticket à la bonne équipe



ARCHITECTURE DU PROJET

- Collecte & Préparation des Données : Extraction, nettoyage, normalisation (Nettoyage, Correction orthographique, Suppression des mots vides, Tokenisation, Lemmatisation, Vectorisation).
- Modèle de Classification : Entraînement et optimisation de l'IA (SVM, Random Forest, Naïve Bayes, etc. ou des modèles plus avancés comme BERT).
- API Backend (Python/Flask/FastAPI) : Point de liaison entre le modèle, la base de données et l'interface.
- Base de Données (Firestore/Firebase) : Stockage des tickets, prédictions, corrections pour l'apprentissage continu.
- Interface Utilisateur (HTML/css/js) : Formulaire de soumission, affichage des prédictions, correction manuelle, historique.
- Dashboard Statistiques (Matplotlib/Seaborn) : Suivi des performances et de l'utilisation.

AVANCÉES RÉALISÉES DURANT LE SPRINT 4

Voici les tâches concrètement menées durant ce sprint :

- Nettoyage et préparation du dataset : suppression des doublons, mise en forme du texte, suppression des stopwords, lemmatisation.
- Entraînement d'un modèle de classification (Random Forest / SVM / autre).
- Tests du modèle : prédiction sur jeux de tests, évaluation (précision, rappel, F1-score).
- Début de l'intégration dans l'interface : création d'un formulaire de soumission de ticket + affichage du résultat de la prédiction.
- Création de scripts Python pour l'analyse automatique.

A-DIFFICULTÉS RENCONTRÉES

- Nettoyage des données texte : difficulté à bien standardiser les descriptions clients (phrases mal formulées, fautes...).
- Choix du bon algorithme : certains modèles donnaient des résultats instables ou trop lents à entraîner.
- Déséquilibre des classes : certaines catégories étaient surreprésentées dans les tickets, ce qui fausse la prédiction.

B-Difficultés d'équipe

- Retards de certaines tâches prévues
- Manque de disponibilité de certains membres à des moments clés
- Communication parfois dispersée (besoin de mieux synchroniser)





BILAN DU SPRINT

1. Ce qui fonctionne bien :

- Modèle fonctionnel avec des résultats encourageants
- Intégration partielle réussie
- Bonne gestion des données

2. Ce qu'il reste à faire :

- Finaliser l'interface utilisateur complète
- Ajouter la supervision des résultats + tableaux statistiques
- Implémenter l'authentification et la gestion des droits
- Améliorer les performances sur des tickets "ambigus"

MERCI