# Sugestão 1

# Chatbot de suporte inteligente com reconhecimento de intenção e extração de entidades (FAQ avançado)

**Problema:** Muitas empresas e instituições recebem um grande volume de perguntas repetitivas. Neste caso, o objetivo é criar um chatbot capaz de entender a pergunta do usuário, identificar sua intenção e extrair informações cruciais (entidades) para fornecer uma resposta precisa e contextualizada, reduzindo a carga sobre o suporte de humanos.

### Componente de Machine Learning:

- Reconhecimento de Intenção (Intent Recognition): Utilizar algoritmos de Classificação de Texto (ex: Naive Bayes, SVM, Redes Neurais com Word Embeddings ou até mesmo Transformers como BERT) para classificar a pergunta do usuário em uma intenção pré-definida. Ex: "Agendamento de Consulta", "Informação de Horário", "Problema com Pedido".
- Extração de Entidades (Named Entity Recognition NER): Usar modelos de sequência ou regras gramaticais/regex para extrair informações específicas da frase do usuário que são cruciais para a resposta. Ex:"data", "horário", "código do produto", "departamento".

## **Componente de Chatbot:**

- Gestão de Diálogo: O chatbot precisará de uma lógica para gerenciar a conversa. Com base na intenção e nas entidades extraídas, ele formulará a resposta, fará perguntas de clarificação se as entidades estiverem faltando, ou encaminhará para um humano quando a confiança for baixa.
- Base de Conhecimento: Integrar o chatbot com uma base de dados de respostas (um FAQ, por exemplo) para que ele possa buscar e apresentar a informação relevante após a interpretação do ML.

#### Tecnologias/Bibliotecas Sugeridas:

- Frameworks de PLN/Chatbot: Rasa Open Source, Dialogflow ES/CX.
- ML: scikit-learn, TensorFlow/Keras ou PyTorch, Hugging Face Transformers.
- Linguagem: Python.

**Resultados Esperados:** Um chatbot que demonstra a capacidade de entender perguntas complexas, extrair dados relevantes e fornecer respostas personalizadas, mostrando o poder da PLN na automatização do suporte ao cliente.

# Sugestão 2

# Assistente de feedback de usuário com análise de sentimento e classificação de Tópicos

**Problema:** Empresas frequentemente recebem feedback de clientes em texto livre, o que dificulta a categorização e a priorização. O projeto visa construir um chatbot que coletar feedback e, através de ML, categorize-o por sentimento (positivo, negativo, neutro) e por tópico (financeiro, técnico, produto, etc.), permitindo uma análise mais rápida e eficiente.

#### Componente de Machine Learning:

- Análise de Sentimento: Desenvolver um modelo de Classificação de Texto para determinar a polaridade ou até a intensidade emocional do feedback do usuário. Pode ser um modelo binário (positivo/negativo), multiclasse (positivo/neutro/negativo), ou com graduações (muito positivo, ligeiramente negativo, etc.).
- Classificação de Tópicos: Outro modelo de Classificação de Texto para categorizar o feedback em diferentes áreas ou departamentos da empresa (ex: "suporte", "vendas", "faturamento", "problemas técnicos", "sugestões de produto").

# **Componente de Chatbot:**

- Coleta Estruturada de Feedback: O chatbot pode guiar o usuário na coleta de feedback, garantindo que as informações necessárias sejam obtidas.
- Feedback ao Usuário/Ações: Após a análise do ML, o chatbot pode dar um feedback ao usuário ("Entendi que você está com um problema técnico e se sentindo frustrado. Vou encaminhar sua solicitação ao nosso time de suporte.") ou registrar o feedback no sistema correto.
- Visualização Interna: Opcionalmente, pode-se pensar em um painel simples que visualize os resultados da classificação de sentimento e tópico dos feedbacks coletados pelo chatbot.

#### Tecnologias/Bibliotecas Sugeridas:

- PLN: NLTK, spaCy, TextBlob (para análise de sentimento inicial).
- ML: scikit-learn, TensorFlow/Keras ou PyTorch (para modelos mais complexos como RNNs ou Transformers para classificação).
- Chatbot: Frameworks como Rasa, ou implementação de um "mini-framework" de chatbot em Flask/Django para gerenciar a conversação.
- Visualização (Opcional): Matplotlib, Seaborn, Dash/Streamlit.
- Linguagem: Python.

**Resultados Esperados:** Um sistema que demonstra como a IA pode processar e organizar feedback textual em larga escala, convertendo dados brutos em insights acionáveis para melhoria de produto/serviço e experiência do cliente.

# Sugestão 3

# Chatbot consultor de conteúdo/produto com sistema de recomendação simples

**Problema:** Em um vasto catálogo de produtos, filmes, livros ou artigos, os usuários muitas vezes se sentem perdidos. Um chatbot que atua como um "consultor" interativo, fazendo perguntas e recomendando itens com base nas preferências expressas pelo usuário, pode enriquecer a experiência de descoberta.

### **Componente de Machine Learning:**

- Sistema de Recomendação:
  - Baseado em Conteúdo (Content-Based): Uma abordagem inicial pode ser recomendar itens com base na similaridade entre as características do item (descrição, categorias, tags) e as preferências declaradas pelo usuário (ML: usar Cosine Similarity entre vetores de texto TF-IDF ou Word Embeddings).
- Processamento de Linguagem Natural (PLN): Para processar as descrições dos itens e as preferências do usuário para o sistema de recomendação.

#### **Componente de Chatbot:**

- Coleta de Preferências: O chatbot inicia um diálogo para entender as preferências do usuário através de perguntas interativas (ex: "Você prefere filmes de ação ou drama?", "Qual o seu estilo de música favorito?").
- Integração com Recomendações: As respostas do usuário são usadas como entrada para o modelo de recomendação. O chatbot então apresenta as recomendações mais relevantes.
- **Refinamento:** O usuário pode dar feedback sobre as recomendações ("Não gostei de X", "Procuro algo mais como Y"), permitindo que o chatbot refina as próximas sugestões.

#### Tecnologias/Bibliotecas Sugeridas:

- ML: scikit-learn (para TF-IDF, Cosine Similarity, etc.), Pandas/NumPy (para manipulação de dados).
- PLN: NLTK, spaCy.

- Chatbot: Rasa (ideal para multi-turn dialogues), ou um chatbot customizado em Flask/Django com um banco de dados simples para armazenar informações dos produtos/conteúdos.
- **Linguagem:** Python.

**Resultados Esperados:** Um chatbot interativo que não apenas responde a perguntas, mas também proativamente auxilia o usuário na descoberta de novos itens, demonstrando a fusão de interação conversacional com algoritmos de recomendação para personalização.