

AI SPAM CLASSIFIER

AI INNOVATION

Machine learning advances: Improved machine learning algorithms, like deep learning and recurrent neural networks, have enhanced the accuracy of spam classifiers by allowing them to learn complex patterns in text data.

Behavioural Analysis: Some spam classifiers now consider the behaviour of users , such as how they interact with emails or messages, to identify spammy patterns and adapt to new spam tactics.

Ensemble model: Combining multiple machine learning models has become common to boost accuracy and reduce false positive in spam classification.

Deep learning for audio and image spam: The use of deep learning for image and audio spam detection has become more prevalent as spammers diversify their methods.

Language models: Advanced language model like GPT-3 can assist in contextual understanding and identifying spam with more natural language processing.

TECH STACK

Front-End

Implementing UI/UX using HTML, CSS, and JavaScript to create an intuitive and user-friendly interface.

The frontend development is responsible for creating the visual interface through which guests interact with the chatbot. It focuses

on delivering a user experience (UX) that is intuitive, visually appealing, and responsive to user actions.

HTML (HyperText Markup Language): HTML forms the structural foundation of the chatbot's interface. It defines the layout and arrangement of elements, allowing for proper organization and presentation of information

CSS (Cascading Style Sheets): CSS is used to style and design the HTML elements. It determines the visual aspects such as colors, fonts, spacing, and layout, ensuring a cohesive and aesthetically pleasing user interface.

BACK END

Python: Python is one of the most popular programming languages for machine learning. You'll need Python installed, and you can use package managers like **pip** to install the necessary libraries.

Flask: Flask is a micro web framework for python that is designed to be lightweight and easy to build web applications and APIs. Its simplicity and extensibility make it a popular option for many developers.

NumPy: NumPy is a fundamental library for numerical and matrix operations. It provides support for arrays and mathematical functions essential for machine learning.

Pandas: Pandas is used for data manipulation and analysis. It's especially helpful for data preprocessing and cleaning.

Scikit-Learn: Scikit-Learn is a powerful library for machine learning. It provides various tools for classification, regression, clustering, and more.

AI SPAM CLASSIFIER

Matplotlib and Seaborn: These libraries are used for data visualization. They are essential for understanding your data and model performance.

TensorFlow or PyTorch: Depending on your preference, you may use TensorFlow or PyTorch as deep learning frameworks. These libraries are essential for building and training neural networks.

Keras: Keras is an abstraction layer that sits on top of TensorFlow and other backends. It simplifies the process of building and training neural networks.

Jupyter Notebook: While not strictly necessary, Jupyter Notebook is a popular tool for creating and sharing documents that contain live code, equations, visualizations, and narrative text. It's often used for developing and documenting machine learning models.

Scipy: Scipy builds on NumPy and provides additional functionality for scientific and technical computing. It includes functions for optimization, integration, interpolation, and more.

XGBoost or LightGBM: These are popular libraries for gradient boosting, which can be particularly useful for structured data and tabular data problems.

NLTK (Natural Language Toolkit) or SpaCy: If your project involves natural language processing (NLP), you'll need one of these libraries to work with text data.

OpenCV: If you're working with computer vision tasks, OpenCV is a valuable package for image and video analysis.

TextBlob: TextBlob is a simple library for processing textual data, including text classification tasks.

Gunicorn (optional): Gunicorn is a production-ready WSGI HTTP server that can be used to serve your Flask application.

Heroku (optional): If you plan to deploy your Flask application to the web, you might want to consider using a cloud platform like Heroku. You'll need to install the Heroku CLI and set up your account.