

SPAM or Legit Classification

1-2 Hours (Estimated) - Technical Test

This is a short technical task, followed by a screen share to discuss what you did.

High level Requirements: 1-2 Hours Estimated

You are required to build a complete machine learning pipeline to classify SMS messages as either spam or ham (legit) using the SMS Spam Collection Dataset [spam.csv](https://www.cs.cmu.edu/~harrylab/spam.csv). After training the model, integrate it into a Django Rest Framework API. The main focus is on feature extraction, avoiding overfitting, and creating a functional backend system.

All code, including the machine learning model and Django API, should be pushed to a **public or private GitHub repository** (provide access if private).

Task Description:

Data Preprocessing:

- Load and preprocess the dataset [spam.csv](https://www.cs.cmu.edu/~harrylab/spam.csv).
- Extract relevant features from the text to be used in model training.
- Split the dataset into training and test sets to evaluate the model.

Model Training:

- Train a model to classify SMS messages as "spam" or "ham (legit)."
- Ensure that the model generalises well by using appropriate validation techniques.
- Evaluate the model using performance metrics such as accuracy, precision, and recall.

API Integration:

- Create a Django Rest Framework API with an endpoint (/predict/) that accepts SMS messages via POST requests in JSON format.
- Load the trained model and use it to classify input SMS messages as "spam" or "ham."
- Implement error handling for invalid inputs.

Bonus Tasks:

Frontend Implementation:

- Create a simple frontend that allows users to input an SMS message and receive a classification ("spam" or "ham") by calling the API.
- The frontend should send the input to the API and display the result in the browser. You can use any frontend framework (e.g., React).

Dockerization:

- Dockerize the entire Django API for smooth Deployment.

Skills Tested:

- AI Proficiency: Familiarity with using AI models and understanding of natural language processing (NLP) concepts.
- API Development: Ability to build simple APIs that integrate AI functionality.
- Problem-Solving: Approach to breaking down tasks, understanding libraries, and applying AI tools to real-world problems.