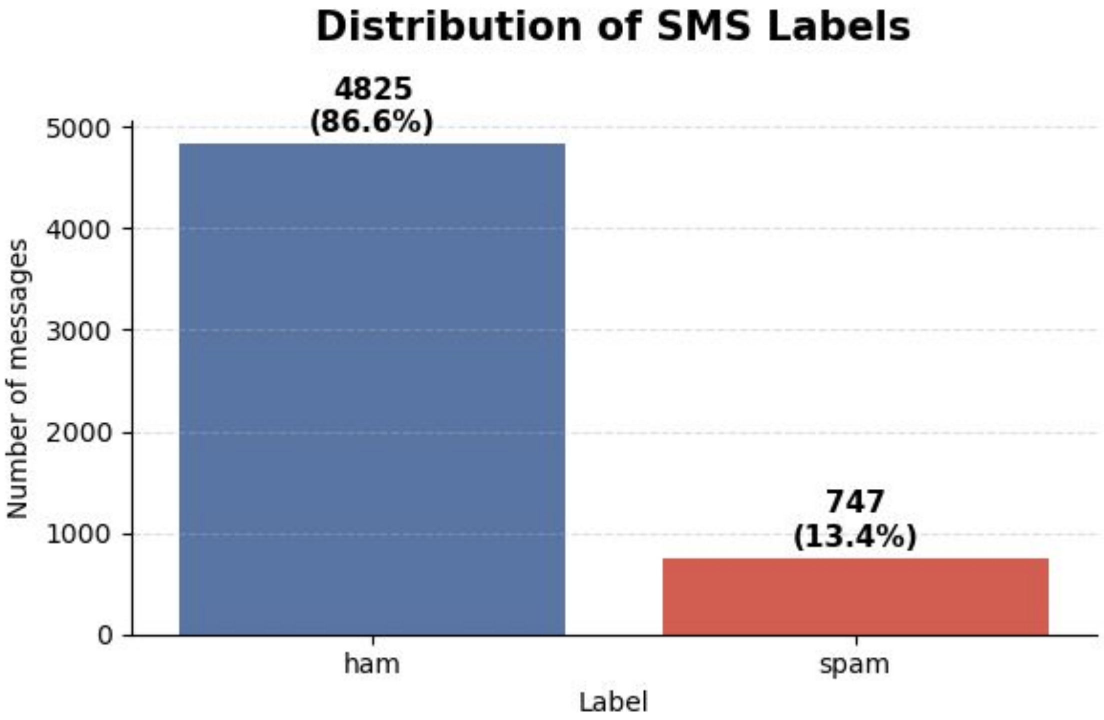
Spam Detector

Automatically classify SMS messages as spam or ham

Project Overview

5572 messages



Data Cleaning & Preprocessing

- **Cleaning:**
  - lower casing
  - removing punctuation
  - keeping digits
- **Tokenization:**
  - split text into words,
  - padded/truncated to 40 tokens
- **Vocabulary:**
  - 7 600 unique words

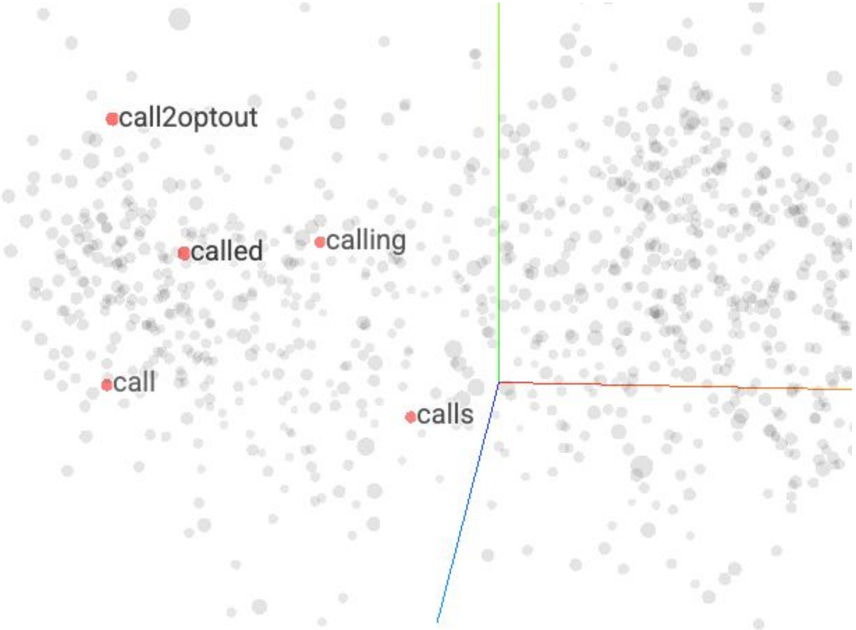
Model Architecture

- *Embedding(128)* → learns semantic meaning of words
- *LSTM(64)* → captures the sequence of words
- *Dropout(0.3)* → reduces overfitting
- *Dense(1, sigmoid)* → outputs spam probability

Layer (type)	Output Shape
text_vectorization_4 (TextVectorization)	(1, 40)
embedding (Embedding)	?
lstm_4 (LSTM)	?
dropout_4 (Dropout)	?
dense_4 (Dense)	?

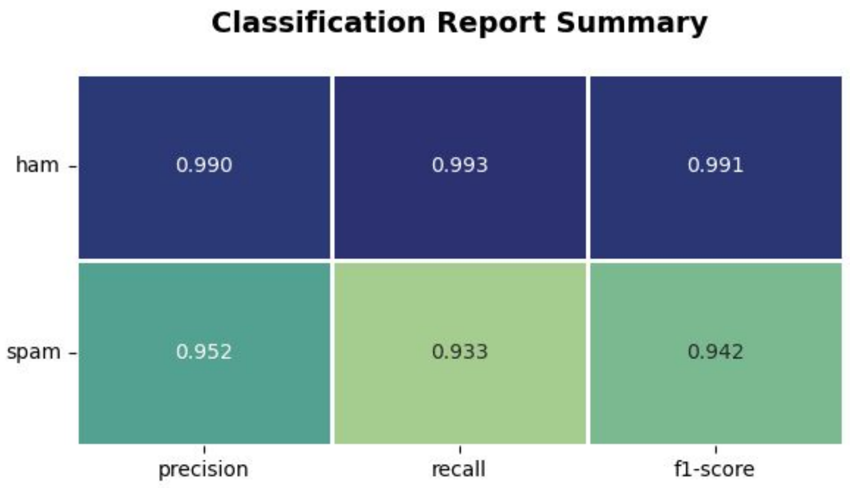
Embedding Visualization

❖ We export the learned word embeddings and visualiz them in TensorFlow Projector:
Similar words cluster together.



Training Results: heatmap

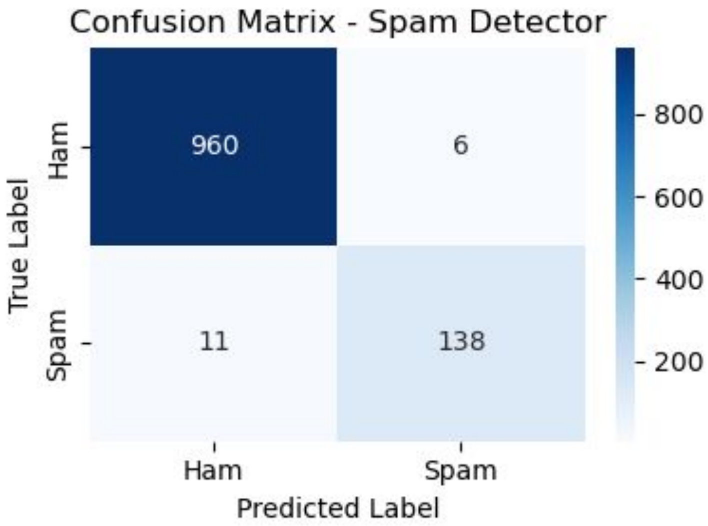
Each cell shows how well the model performs for **ham** and **spam** in terms of **precision**, **recall**, and **F1-score**.



F1-score combines **precision** and **recall** – near-perfect values indicate a robust model.

Training Results: Confusion matrix

➢ Each cell shows the number of messages falling into a specific category of prediction vs reality.



True Negative Real *ham* predicted as *ham*	**False Positive** Real *ham* predicted as *spam*
False Negative Real *spam* predicted as *ham*	**True Positive** Real *spam* predicted as *spam*

Error Analysis

False Positives
(predicted spam but actually ham):

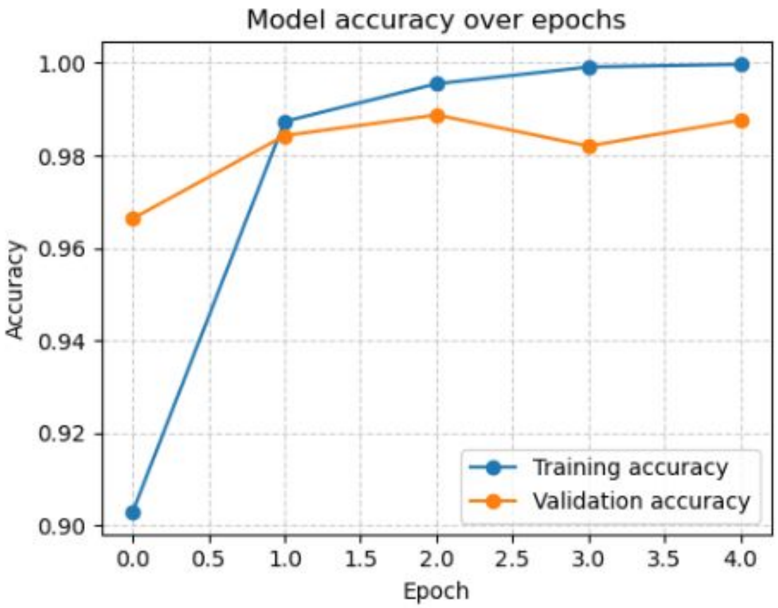
- [0.87] waiting for your call
- [0.81] nokia phone is lovly
- [0.81] height of confidence all the aeronautics professors wer calld amp they wer askd 2 sit in an aeroplane aftr they sat they...
- [0.52] unlimited texts limited minutes

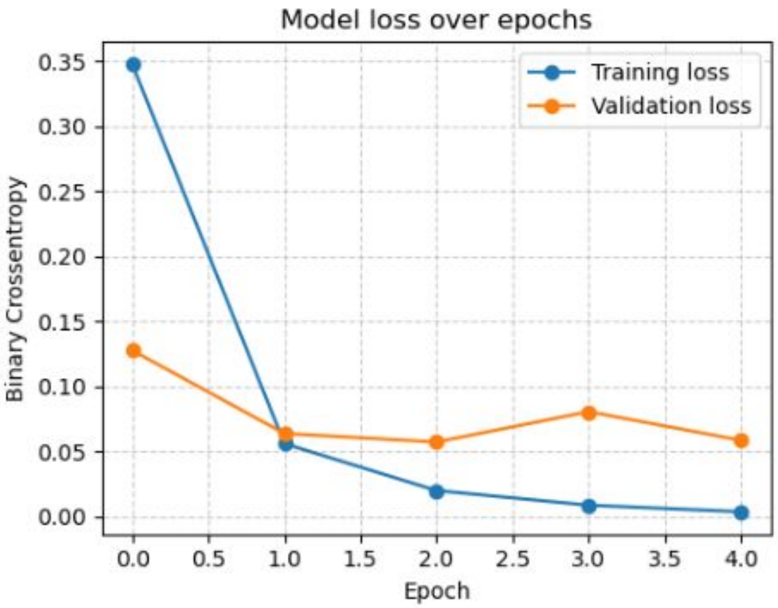
False Negatives
(predicted ham but actually spam):

- [0.00] sorry i missed your call let's talk when you have the time i'm on 07090201529
- [0.00] for sale arsenal dartboard good condition but no doubles or trebles
- [0.18] latest news police station toilet stolen cops have nothing to go on

True Negative Real *ham* predicted as *ham*	**False Positive** Real *ham* predicted as *spam*
False Negative Real *spam* predicted as *ham*	**True Positive** Real *spam* predicted as *spam*

Training Curves: Accuracy and Loss over Epochs





Training and validation curves converge – stable learning, no overfitting.

Prediction Tool: Real-time Spam Detection

This simple prediction interface allows testing the model with new unseen messages.

- ➢ The tool takes raw text input (SMS) and predicts its **probability of being spam**.
- ➢ Messages are color-coded: **green for HAM**, **red for SPAM**.

	Message	Predicted Label	Spam Probability
0	Congratulations! You've won a new iPhone, click here to claim!	SPAM	94.97%
1	Hi John, can you send me the report by tomorrow?	HAM	0.06%
2	Urgent! Your bank account has been locked, verify immediately.	SPAM	90.82%
3	Ok cool, I'll bring the cake for Saturday.	HAM	0.21%

Thank you for your attention
– any questions?

Model Comparison & Choice Justification

Model	Strengths	Limitations	Relevance for SMS Spam
Naive Bayes / Logistic Regression	Fast, simple, good baseline	No context understanding	✅ Useful as a baseline
CNN (Convolutional Neural Network)	Detects local word patterns like "free offer"	Misses long-term dependencies	⚪ Good alternative
LSTM (Long Short-Term Memory)	Captures sequence and context	Slightly slower to train	🟢 **Best fit for SMS**
Transformer (BERT, DistilBERT)	Powerful semantic understanding	Heavy, overkill for small dataset	🔵 For future exploration

Possible improvements

1️⃣**Try hybrid architectures (CNN + LSTM)**
→ Combine the local feature detection of CNNs with the contextual understanding of LSTMs.
This could enhance accuracy on tricky “soft spam” messages.

2️⃣**Experiment with Transformer embeddings (BERT / DistilBERT)**
→ Pretrained transformers can capture subtle semantic nuances, improving generalization on unseen messages.

3️⃣**Adjust classification threshold**
→ Fine-tuning the spam probability cutoff (e.g. 0.4 → 0.6) can optimize precision vs. recall depending on business needs.

4️⃣**Deploy as a real-time API**
→ The model could be integrated into a customer SMS filtering system or chatbot moderation tool.