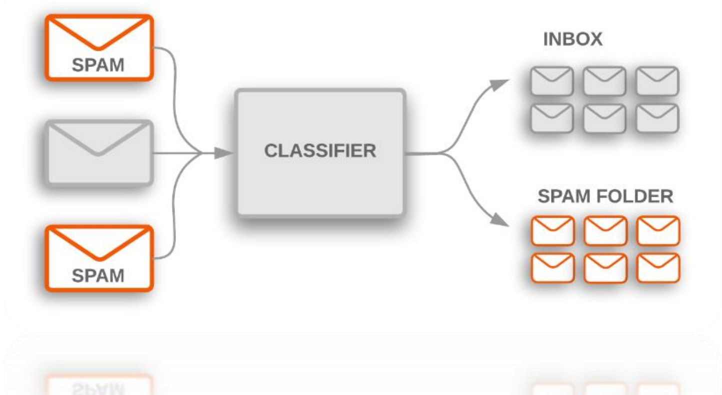
Phase 1: Problem Definition and Design Thinking

PROJECT TITLE	AI SPAM CLASSIFIER
NAME	SAKTHI VELU S
TEAM ID	5232 Prj_204227_Team_1
TEAM NAME	
COLLEGE CODE-NAME	9238-MANGAYARKARASI COLLEGE OF ENGINEERING PARAVAI, MADURAI.
GROUP	5
GITHUB REPOSITORY LINK	https://github.com/Sakthi0604/IBM-NAAN-MUDHALVAN-AI.git

AI SPAM CLASSIFIER



1. PROJECT DEFINITION:

1.1. PROJECT OBJECTIVES:

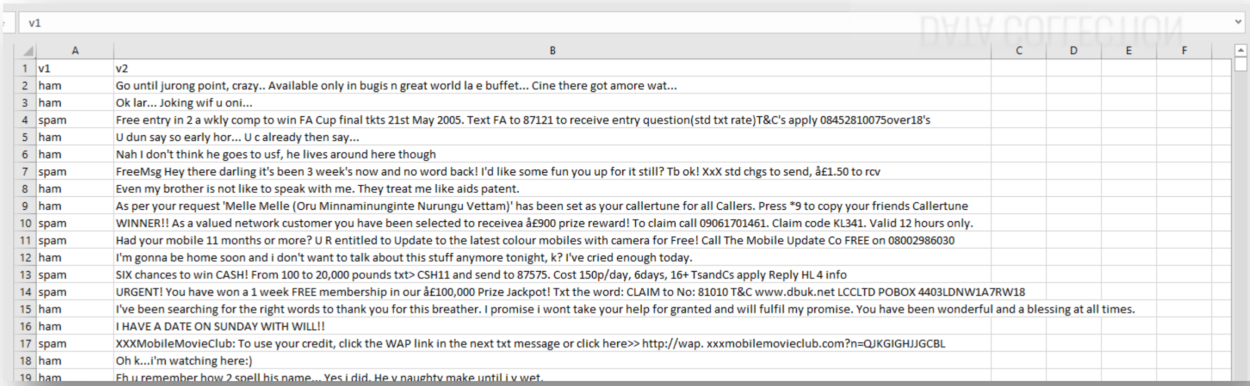
Our project aims to create a highly effective AI-powered spam classifier, capable of accurately differentiating between spam and non-spam messages in emails or text messages. The primary objective is to minimize both false positives (wrongly identifying legitimate messages as spam) and false negatives (missing actual spam messages), while maintaining a high level of classification accuracy.

2. DESIGN THINKING:

To address this problem effectively, we will follow a structured approach involving the following key steps:

2.1. DATA COLLECTION:

Data collection is the first step in building our spam classifier. In this phase, we need to gather a dataset that contains examples of both spam and non-spam (ham) messages. The dataset linked from Kaggle (SMS Spam Collection Dataset) will serve as our source of labelled data. This dataset likely consists of text messages, each labelled as either spam or non-spam.



	A	B	C	D	E	F
1	v1	v2				
2	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...				
3	ham	Ok lar... Joking wif u oni...				
4	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's				
5	ham	U dun say so early hor... U c already then say...				
6	ham	Nah I don't think he goes to usf, he lives around here though				
7	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, â£1.50 to rcv				
8	ham	Even my brother is not like to speak with me. They treat me like aids patent.				
9	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune				
10	spam	WINNER!! As a valued network customer you have been selected to receivea â£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.				
11	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030				
12	ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.				
13	spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info				
14	spam	URGENT! You have won a 1 week FREE membership in our â£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18				
15	ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.				
16	ham	I HAVE A DATE ON SUNDAY WITH WILL!!				
17	spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL				
18	ham	Oh k...i'm watching here:)				
19	ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.				

2.2. DATA PREPROCESSING

Before feeding the text data into our machine learning model, we need to prepare it. Data preprocessing includes the following key steps:

Text Cleaning: We clean the text data by removing any special characters, HTML tags, or other noisy elements that may not contribute to the classification task. This step ensures that the text data is in a more standardized format.

```
import re
cleaned_text = re.sub(r"<.*?>", "", text)  # Remove HTML tags
```

Lowercasing: Converting all the text to lowercase helps in achieving consistency. It ensures that the model treats "Spam" and "spam" as the same word, reducing ambiguity.

```
lowercased_text = text.lower()
```

Tokenization: Tokenization is the process of splitting the text into individual words or tokens. This step breaks down the text into its smallest meaningful units, making it easier for the model to work with.

```
import nltk

from nltk.tokenize import word_tokenize

tokens = word_tokenize(text)
```

2.3. FEATURE EXTRACTION:

2.4. Machine learning algorithms typically work with numerical data, so we need to convert our text data into numerical features. To achieve this, we will use the TF-IDF (Term Frequency-Inverse Document Frequency) technique:

TF-IDF: TF-IDF assigns a numerical value to each word in the text based on its frequency within a specific message (Term Frequency) and its importance across the entire dataset (Inverse Document Frequency). This creates a numerical representation of each message, where words that are common in a specific message but rare in the dataset receive higher values.

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Create a TfidfVectorizer object

tfidf_vectorizer = TfidfVectorizer()

# Fit and transform the documents to TF-IDF vectors

tfidf_matrix = tfidf_vectorizer.fit_transform(documents)

# Get the feature names (words) corresponding to the columns in the TF-IDF matrix

feature_names = tfidf_vectorizer.get_feature_names_out()

# Convert the TF-IDF matrix to a dense array (optional)

tfidf_matrix_dense = tfidf_matrix.toarray()

# Create a dictionary to store TF-IDF values for each document

tfidf_results = {}
```

2.5. MODEL SELECTION:

Choosing the right machine learning algorithm is crucial for building an effective spam classifier. We explore various options:

1. Naive Bayes: This probabilistic algorithm is suitable for text classification tasks. It calculates the probability of a message being spam or non-spam based on word frequencies.

```
from sklearn.naive_bayes import MultinomialNB
classifier = MultinomialNB()
classifier.fit(X_train, y_train)
```

2. Support Vector Machines (SVM): SVMs are effective for linear and nonlinear classification tasks. They aim to find a decision boundary that best separates spam from non-spam messages.

```
from sklearn.svm import SVC
classifier = SVC(kernel='linear')
classifier.fit(X_train, y_train)
```

3. Deep Learning: Deep learning, particularly using neural networks, offers a more complex approach to text classification. It can automatically learn intricate patterns in the data, potentially leading to high accuracy.

```
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from sklearn.model_selection import train_test_split
model = Sequential()
model.add(Embedding(max_words, 128, input_length=max_sequence_length))
model.add(LSTM(128, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(1, activation='sigmoid'))

# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam',
metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=5, batch_size=32)
```

2.6. EVALUATION

The performance of the developed model will undergo a rigorous evaluation using pertinent evaluation metrics, including:

- **Accuracy**: Quantifying the proportion of correctly predicted cases.

```
from sklearn.metrics import accuracy_score, classification_report
accuracy = accuracy_score(y_test, y_pred)
```

- **Precision**: Assessing the model's capability to correctly identify individuals with diabetes among those predicted to have it.

```
From sklearn.metrics import precision_score
precision = precision_score(true_labels, predicted_labels)
```

- **Recall**: Gauging the model's ability to identify all individuals with diabetes within the dataset.

```
from sklearn.metrics import recall_score
recall = recall_score(true_labels, predicted_labels)
```

- **F1-Score**: Determining the harmonic mean of precision and recall, offering a balanced assessment.

```
from sklearn.metrics import f1_score
f1 = f1_score(true_labels, predicted_labels)
```

2.7. Iterative Improvement

Building a robust spam classifier may require refining and optimizing the initial model. Some strategies for iterative improvement include:

Hyperparameter Tuning: Experimenting with different model parameters to find the settings that optimize performance.

Feature Engineering: Exploring additional features or advanced text preprocessing techniques to enhance the model's understanding of the data.

Regularization: Implementing techniques like dropout or L2 regularization to prevent overfitting, where the model performs well on training data but poorly on new data.

Ensemble Methods: Combining multiple models (e.g., using ensemble techniques like bagging or boosting) to improve overall accuracy and reliability.

Portions that are well-explained In this section, we'll talk about concentrating more on the major findings and conclusions of the research Supervised machine learning has a high acceptance rate. Throughout the review, the approach can be noticed. This strategy is effective. is employed primarily because it produces more accurate findings. With less fluctuation, this strategy has a high level of consistency. Aside from that, we've discovered that certain algorithms work better than others. When compared to other techniques, such as Nave Based and SVM, there is a strong demand for them.