

Building a Smarter AI-Powered Spam Classifier Phase 3

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm

spam = pd.read_csv('C:\\Users\\nethm\\Downloads\\spam.csv')
z = spam['EmailText']
y = spam["Label"]
z_train, z_test, y_train, y_test = train_test_split(
    z, y, test_size = 0.2)

cv = CountVectorizer()
features = cv.fit_transform(z_train)

model = svm.SVC()
model.fit(features, y_train)

features_test = cv.transform(z_test)
print(model.score(features_test, y_test))
```

Select model:

Choose the appropriate machine learning or deep learning model for classification. Popular choices include Naive Bayes, Support Vector Machines, Random Forests, and neural networks like LSTM or CNN.

Training:

Split the dataset into training set and validation set. Train the selected model on the training data and fine-tune the hyperparameters to optimize performance.

Review:

Evaluate model performance using metrics such as accuracy, precision, recall, F1 score, and ROC-AUC. Adjust models and features based on evaluation results.

Overall method:

Consider using aggregation methods such as stacking or boosting to improve classification accuracy.

Cross validation:

Perform cross-validation to ensure generalizability of the model.

Real-time scoring:

Deploy the trained model in a real-time environment where the model can classify incoming emails or messages as spam or not.

Feedback loop:

Continuously monitor the classifier's performance and periodically retrain it with new data to adapt to changing spam patterns.

