EMAIL SPAM DETECTION WITH MACHINE LEARNING Import the necessary dependencies. Pandas is a library used mostly used by data scientists for data cleaning and analysis.

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

from google.colab import drive
drive.mount("/content/gdrive")
```

```
Mounted at /content/gdrive
```

Now we'll create a spam.csv file which we'll turn into a data frame and save to our folder spam.

A data frame is a structure that aligns data in a tabular fashion in rows and columns.

```
spam = pd.read_csv('/content/gdrive/MyDrive/OIBSIP/spam.csv',encoding='latin-1')
```

Use train-test split method to train our email spam detector to recognize and categorize spam emails.

We can use it for either classification or regression of any supervised learning algorithm.

Train dataset: used to fit the machine learning model

Test dataset: used to evaluate the fit of the machine learning model

```
z = spam['v2']
y = spam["v1"]
z_train, z_test,y_train, y_test = train_test_split(z,y,test_size = 0.2)
```

Extracting the features

CountVectorizer() randomly assigns a number to each word in a process called tokenizing. Then, it counts the number of occurrences of words and saves it to cv.

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

SVM, the support vector machine algorithm, is a linear model for classification and regression.

The idea of SVM is simple, the algorithm creates a line, or a hyperplane, which separates the data into classes. SVM can solve both linear and non-linear problems:

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

```
SVC()
```

In the model.fit(features,y_train) function, model.fit trains the model with features and y_train. Then, it checks the prediction against the y_train label and adjusts its parameters until it reaches the highest possible accuracy.

```
features_test = cv.transform(z_test)
print("Accuracy: {}".format(model.score(features_test,y_test)))
```

```
Accuracy: 0.9820627802690582
```

✓ 0s completed at 22:16