```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score

def load_dataset(file_path):
    data = pd.read_csv(file_path, encoding='latin-1')
    email_text = data['v2']  # Assuming 'v2' is the column name containing the email text
    labels = data['v1']  # Assuming 'v1' is the column name containing the spam or non-spa
    return email_text, labels

def preprocess_data(email_text, labels):
    tfidf = TfidfVectorizer()
    X = tfidf.fit_transform(email_text)
    y = labels
    return X, y, tfidf

def build_model(X, y):
    model = SVC()
    model.fit(X, y)
    return model

def evaluate_model(model, X_test, y_test):
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    return accuracy

def predict_new_emails(model, new_emails, tfidf):
    new_emails_transformed = tfidf.transform(new_emails)
    new_emails_pred = model.predict(new_emails_transformed)
    return new_emails_pred

# Step 1: Load and preprocess the dataset
email_text, labels = load_dataset('spam.csv')
X, y, tfidf = preprocess_data(email_text, labels)

# Step 2: Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 3: Build the machine learning model
model = build_model(X_train, y_train)

# Step 4: Evaluate the model
accuracy = evaluate_model(model, X_test, y_test)
print("Accuracy:", accuracy)

# Step 5: Deploy the Spam Detector (classification of new emails)
new_emails = ["Hey there! meet me soon."]
new_emails_pred = predict_new_emails(model, new_emails, tfidf)
print("New Email Predictions:", new_emails_pred)
print(X_train)
print(y_train)
```

```
Accuracy: 0.9766816143497757
New Email Predictions: ['ham']
  (0, 1260)     0.17167311365375632
```

```
  (0, 1595)     0.38297119137367513
  (0, 1976)     0.27594587972305035
  (0, 3711)     0.27594587972305035
  (0, 3831)     0.22373410370588745
  (0, 4087)     0.1363554359676013
  (0, 4939)     0.14378067065478475
  (0, 5113)     0.3641249694942899
  (0, 5223)     0.14915391012997223
  (0, 5241)     0.3507533622715513
  (0, 5367)     0.1760662062543487
  (0, 5525)     0.15967788028036542
  (0, 5609)     0.18841033141508542
  (0, 6598)     0.26651308736220675
  (0, 7253)     0.2129859169630051
  (0, 7627)     0.1250021552728397
  (0, 8604)     0.28349984119511407
  (1, 0)        0.24008719095129516
  (1, 483)      0.26022133591777746
  (1, 1380)     0.23453811904688157
  (1, 1783)     0.1821151920149335
  (1, 1914)     0.2832457567394777
  (1, 3070)     0.2832457567394777
  (1, 3308)     0.10616120844121778
  (1, 3622)     0.2832457567394777
  :     :
  (4452, 5894)  0.33817840131896065
  (4452, 6951)  0.28982640366310375
  (4452, 7756)  0.11377820288146219
  (4452, 8032)  0.20383472085602383
  (4453, 1813)  0.3930074473017961
  (4453, 4525)  0.5612643415409254
  (4453, 4669)  0.4827272885665661
  (4453, 7076)  0.5454373017617413
  (4454, 3387)  0.3789492695016858
  (4454, 3797)  0.31683821458059697
  (4454, 5994)  0.4347949983559448
  (4454, 6281)  0.4093023796257788
  (4454, 7082)  0.4468981250591947
  (4454, 7083)  0.4468981250591947
  (4455, 4318)  0.5614796075096616
  (4455, 5427)  0.47775734583636464
  (4455, 6743)  0.5354025918875037
  (4455, 7774)  0.4121075508290671
  (4456, 3174)  0.44983924830359445
  (4456, 3563)  0.34800538786721863
  (4456, 3770)  0.25933477682815825
  (4456, 4087)  0.4535640915696941
  (4456, 5023)  0.4250652003623859
  (4456, 5420)  0.26652755854790505
  (4456, 8482)  0.38967245866556843
1978     ham
3989    spam
3935     ham
4078     ham
4086    spam
```

Colab paid products - Cancel contracts here