source code github

github: https://github.com/HUyEsona/-ML-project_-Classifying-Spam-Emails.git

Spam Detection with Logistic Regression

Step 1: Nhập thư viện

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
from scipy.sparse import csr_matrix
```

Step 2: Logistic Regression Class (Lớp hồi quy logistic)

Xác định mô hình hồi quy logistic với tối ưu hóa giảm độ dốc.

```
class LogisticRegression:
    def __init__(self, learning_rate=0.01, num_iterations=1000):
        self.learning_rate = learning_rate
        self.num_iterations = num_iterations
        self.weights = None
        self.bias = None
        self.loss_history = []

    def sigmoid(self, z):
        return 1 / (1 + np.exp(-z))

    def compute_loss(self, y_pred, y):
        epsilon = 1e-10
        loss = -np.mean(y * np.log(y_pred + epsilon) + (1 - y) *
np.log(1 - y_pred + epsilon))
        return loss

    def fit(self, X, y):
        num_samples, num_features = X.shape
        if isinstance(X, csr_matrix):
            X = X.toarray()
        self.weights = np.zeros(num_features)
        self.bias = 0
        for _ in range(self.num_iterations):
            linear_model = np.dot(X, self.weights) + self.bias
            y_pred = self.sigmoid(linear_model)
            loss = self.compute_loss(y_pred, y)
            self.loss_history.append(loss)
            dw = (1 / num_samples) * np.dot(X.T, (y_pred - y))
            db = (1 / num_samples) * np.sum(y_pred - y)
            self.weights -= self.learning_rate * dw
            self.bias -= self.learning_rate * db

    def predict(self, X):
        if isinstance(X, csr_matrix):
            X = X.toarray()
        linear_model = np.dot(X, self.weights) + self.bias
        y_pred = self.sigmoid(linear_model)
        y_pred_class = np.where(y_pred > 0.5, 1, 0)
        return y_pred_class
```

Step 3: Tải và xử lý trước dữ liệu

```
raw_mail_data = pd.read_csv('mail_SPAM_data.csv')

mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)), '')


mail_data.loc[mail_data['Category'] == 'spam', 'Category'] = 0
mail_data.loc[mail_data['Category'] == 'ham', 'Category'] = 1

X = mail_data['Message']
Y = mail_data['Category'].astype(int)
```

Step 4: Tách thử nghiệm đào tạo và trích xuất tính năng

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=5)

feature_extraction = TfidfVectorizer(min_df=1, stop_words='english',
lowercase=True)
X_train_feature = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)
```

Step 5: Huấn luyện mô hình hồi quy logistic

```
model = LogisticRegression()
model.fit(X_train_feature, Y_train)
```

Step 6: DỰ ĐOÁN TIN NHẮN

```
file_name = input("Nhập path file CSV chứa dữ liệu cần dự đoán: ")

new_mail_data = pd.read_csv(file_name)
new_messages = new_mail_data['Message']
new_messages_features = feature_extraction.transform(new_messages)

new_predictions = model.predict(new_messages_features)

new_mail_data['Prediction'] = new_predictions
new_mail_data['Prediction'] = new_mail_data['Prediction'].map({1: 
'Ham', 0: 'Spam'})

print(new_mail_data[['Message', 'Prediction']])
```

```
                                                Message Prediction
0     Go until jurong point, crazy.. Available only ...        Ham
1                         Ok lar... Joking wif u oni...        Ham
2     Free entry in 2 a wkly comp to win FA Cup fina...        Ham
3     U dun say so early hor... U c already then say...        Ham
4     Nah I don't think he goes to usf, he lives aro...        Ham
...                                                 ...        ...
5567  This is the 2nd time we have tried 2 contact u...        Ham
5568               Will ü b going to esplanade fr home?        Ham
5569  Pity, * was in mood for that. So...any other s...        Ham
5570  The guy did some bitching but I acted like i'd...        Ham
5571                         Rofl. Its true to its name        Ham

[5572 rows x 2 columns]
```

kết quả in ra