AI-spam_classifier

Preprocessing data

```
import numpy as np

import pandas as pd

df = pd.read_csv('C:\Users\dhanasree\AI-spam_classifier\AI-spam_classifier\spam_ham_dataset.csv')

df.sample(5)

df.shape
```

Data cleaning

```
df.info()

df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)

df.sample(5)

df.rename(columns={'v1':'target','v2':'text'},inplace=True)

df.sample(5)

from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()

df['target'] = encoder.fit_transform(df['target'])

df.head()

df.isnull().sum()

df.duplicated().sum()

df = df.drop_duplicates(keep='first')

df.duplicated().sum()

df.shape
```

EDA

```
df.head()
df['target'].value_counts()
import matplotlib.pyplot as plt
plt.pie(df['target'].value_counts(), labels=['ham','spam'],autopct="%0.2f")
plt.show()
import nltk
nltk.download('punkt')
df['num_characters'] = df['text'].apply(len)
df.head()
df['num_words'] = df['text'].apply(lambda x:len(nltk.word_tokenize(x)))
df.head()
df['num_sentences'] = df['text'].apply(lambda x:len(nltk.sent_tokenize(x)))
df.head()
df[['num_characters','num_words','num_sentences']].describe()
df[df['target'] == 0][['num_characters','num_words','num_sentences']].describe()
df[df['target'] == 1][['num_characters','num_words','num_sentences']].describe()
import seaborn as sns
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_characters'])
sns.histplot(df[df['target'] == 1]['num_characters'],color='red')
plt.figure(figsize=(12,6))
sns.histplot(df[df['target'] == 0]['num_words'])
sns.histplot(df[df['target'] == 1]['num_words'],color='red')
sns.pairplot(df,hue='target')
sns.heatmap(df.corr(),annot=True)
```

Data Preprocessing

```
def transform_text(text):
    text = text.lower()
    text = nltk.word_tokenize(text)


    y = []
    for i in text:
        if i.isalnum():
            y.append(i)
    text = y[:]
    y.clear()


    for i in text:
        if i not in stopwords.words('english') and i not in string.punctuation:
            y.append(i)
    text = y[:]
    y.clear()


    for i in text:
        y.append(ps.stem(i))
    return " ".join(y)
transform_text("I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.")
df['text'][10]
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
ps.stem('loving')
df['transformed_text'] = df['text'].apply(transform_text)
df.head()
```

```
from wordcloud import WordCloud
wc = WordCloud(width=500,height=500,min_font_size=10,background_color='white')
spam_wc = wc.generate(df[df['target'] == 1]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(15,6))
plt.imshow(spam_wc)
ham_wc = wc.generate(df[df['target'] == 0]['transformed_text'].str.cat(sep=" "))
plt.figure(figsize=(15,6))
plt.imshow(ham_wc)
df.head()
spam_corpus = []
for msg in df[df['target'] == 1]['transformed_text'].tolist():
    for word in msg.split():
        spam_corpus.append(word)
len(spam_corpus)
from collections import Counter
sns.barplot(pd.DataFrame(Counter(spam_corpus).most_common(30))[0],pd.DataFrame(Counter(spam_corpus).most_common(30))[1])
plt.xticks(rotation='vertical')
plt.show()
ham_corpus = []
for msg in df[df['target'] == 0]['transformed_text'].tolist():
    for word in msg.split():
        ham_corpus.append(word)
len(ham_corpus)
from collections import Counter
sns.barplot(pd.DataFrame(Counter(ham_corpus).most_common(30))[0],pd.DataFrame(Counter(ham_corpus).most_common(30))[1])
plt.xticks(rotation='vertical')
plt.show()
df.head()
```