Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simulat

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[3]:
```
pip install ipywidgets
```

```
Requirement already satisfied: ipywidgets in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (8.1.8)
Requirement already satisfied: comm>=0.1.3 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipywidgets) (0.2.3)
Requirement already satisfied: ipython>=6.1.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipywidgets) (9.4.0)
Requirement already satisfied: traitlets>=4.3.1 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipywidgets) (5.14.3)
Requirement already satisfied: widgetsnbextension~=4.0.14 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipywidgets)
(4.0.15)
Requirement already satisfied: jupyterlab_widgets~=3.0.15 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipywidgets)
(3.0.16)
Requirement already satisfied: colorama in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.1.0->ipywidgets) (0.
4.6)
Requirement already satisfied: decorator in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.1.0->ipywidgets) (5.
2.1)
Requirement already satisfied: ipython-pygments-lexers in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.1.0->i
pywidgets) (1.1.1)
Requirement already satisfied: jedi>=0.16 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.1.0->ipywidgets)
(0.19.2)
Requirement already satisfied: matplotlib-inline in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.1.0->ipywidg
ets) (0.1.7)
Requirement already satisfied: prompt_toolkit<3.1.0,>=3.0.41 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.
1.0->ipywidgets) (3.0.51)
Requirement already satisfied: pygments>=2.4.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.1.0->ipywidget
s) (2.19.2)
Requirement already satisfied: stack_data in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from ipython>=6.1.0->ipywidgets)
(0.6.3)
Requirement already satisfied: wcwidth in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from prompt_toolkit<3.1.0,>=3.0.41->ip
ython>=6.1.0->ipywidgets) (0.2.13)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from jedi>=0.16->ipython>=
6.1.0->ipywidgets) (0.8.4)
Requirement already satisfied: executing>=1.2.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from stack_data->ipython>=6.
1.0->ipywidgets) (2.2.0)
Requirement already satisfied: asttokens>=2.1.0 in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from stack_data->ipython>=6.
1.0->ipywidgets) (3.0.0)
Requirement already satisfied: pure-eval in c:\users\hp\appdata\local\programs\python\python313\lib\site-packages (from stack_data->ipython>=6.1.0->ipy
widgets) (0.2.3)
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[5]:

```
import pandas as pd
import re
import nltk
import ipywidgets as widgets
from IPython.display import display, clear_output

from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB

nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\HP\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

[5]: True

Load Dataset

[14]:

```
import matplotlib.pyplot as plt
data = pd.read_csv(r"C:\Users\HP\Downloads\spam.csv", encoding="latin-1")
data = data[['v1', 'v2']]
data.columns = ['label', 'message']
data.head()
```

[14]:

	label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

Text Preprocessing

[15]:

```
ps = PorterStemmer()
stop_words = set(stopwords.words('english'))

def preprocess(text):
    text = text.lower()
    text = re.sub('[^a-zA-Z]', ' ', text)
    words = text.split()
    words = [ps.stem(w) for w in words if w not in stop_words]
    return " ".join(words)

data['clean_message'] = data['message'].apply(preprocess)
```

Feature Extraction & Model Training

[16]:

```
tfidf = TfidfVectorizer()
X = tfidf.fit_transform(data['clean_message']).toarray()
y = data['label'].map({'ham': 0, 'spam': 1})

model = MultinomialNB()
model.fit(X, y)
```

[16]:

▾ MultinomialNB

▸ Parameters

[17]:

```
data['length'] = data['message'].apply(len)

plt.hist(data['length'], bins=50)
plt.title("Message Length Distribution")
plt.xlabel("Message Length")
plt.ylabel("Frequency")
plt.show()
```

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

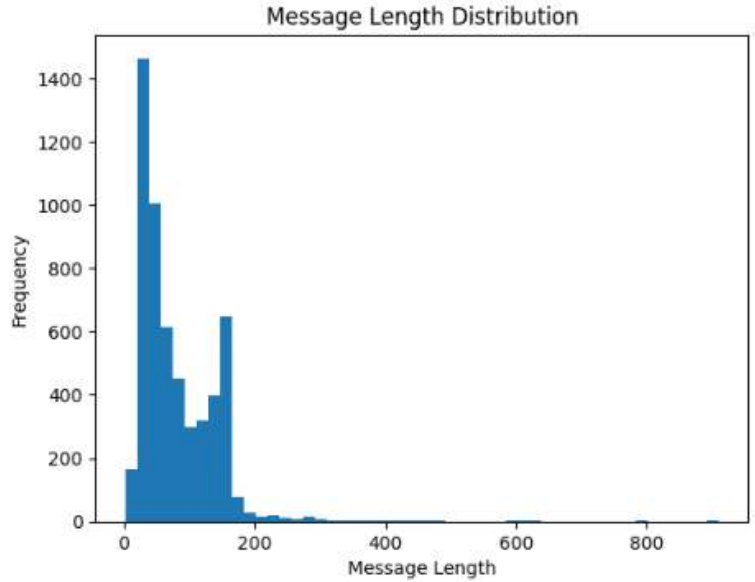
File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[17]:
```
data['length'] = data['message'].apply(len)

plt.hist(data['length'], bins=50)
plt.title("Message Length Distribution")
plt.xlabel("Message Length")
plt.ylabel("Frequency")
plt.show()
```



[18]:
```
spam_len = data[data['label']=='spam']['length']
ham_len = data[data['label']=='ham']['length']

plt.hist(ham_len, bins=50, alpha=0.7, label='Ham')
```

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

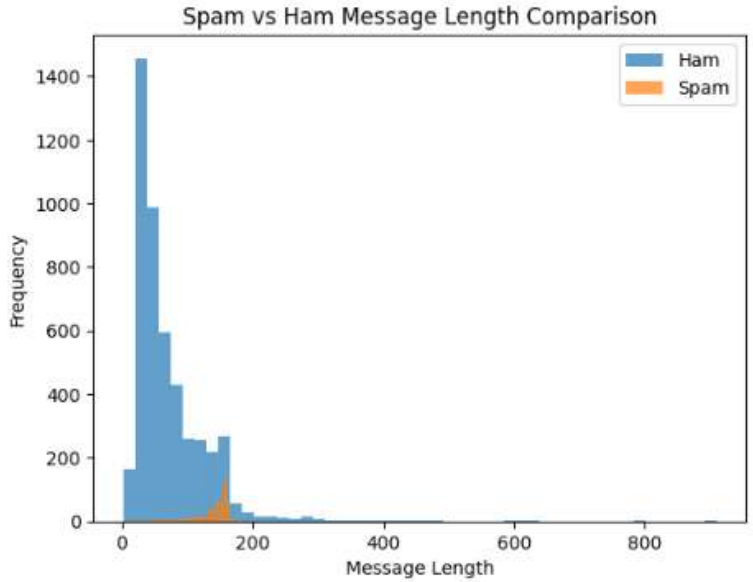
File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

[18]:
```
spam_len = data[data['label']=='spam']['length']
ham_len = data[data['label']=='ham']['length']

plt.hist(ham_len, bins=50, alpha=0.7, label='Ham')
plt.hist(spam_len, bins=50, alpha=0.7, label='Spam')
plt.legend()
plt.title("Spam vs Ham Message Length Comparison")
plt.xlabel("Message Length")
plt.ylabel("Frequency")
plt.show()
```



[19]:
```
from sklearn.model_selection import train_test_split
```

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

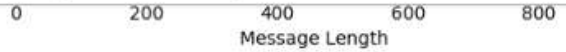
jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Trusted

Code

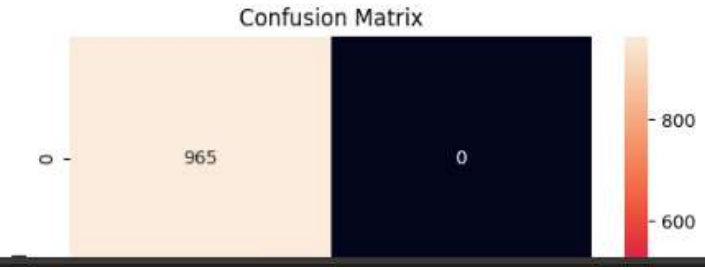
JupyterLab Python 3 (ipykernel)



[19]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

[20]:
```
from sklearn.naive_bayes import MultinomialNB

model = MultinomialNB()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
```

[21]:
```
from sklearn.metrics import confusion_matrix
import seaborn as sns

cm = confusion_matrix(y_test, y_pred)

sns.heatmap(cm, annot=True, fmt='d')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```



Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

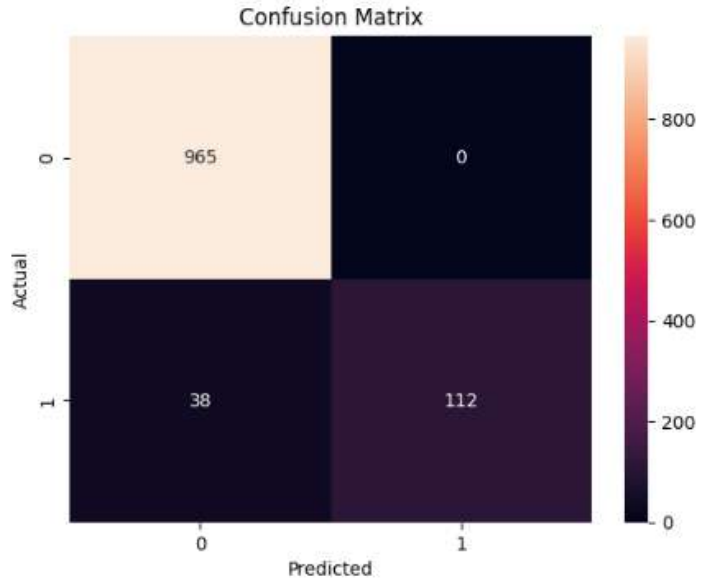
jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Trusted

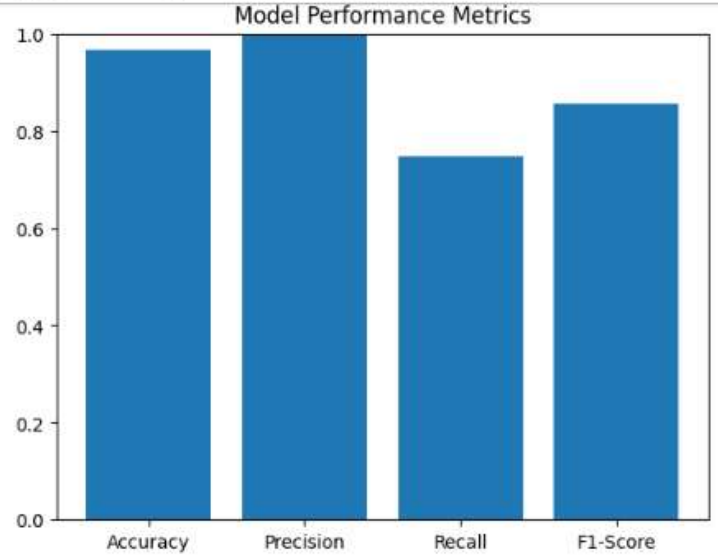
Code

JupyterLab Python 3 (ipykernel)



[22]:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

metrics = {
    "Accuracy": accuracy_score(y_test, y_pred),
    "Precision": precision_score(y_test, y_pred),
    "Recall": recall_score(y_test, y_pred),
    "F1-Score": f1_score(y_test, y_pred)
}

plt.bar(metrics.keys(), metrics.values())
plt.title("Model Performance Metrics")
plt.ylim(0,1)
plt.show()
```



[23]:
```
from sklearn.metrics import roc_curve, auc

y_prob = model.predict_proba(X_test)[:,1]
fpr, tpr, _ = roc_curve(y_test, y_prob)
roc_auc = auc(fpr, tpr)

plt.plot(fpr, tpr, label="AUC = %.2f" % roc_auc)
plt.plot([0,1], [0,1], linestyle='--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
plt.legend()
plt.show()
```

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

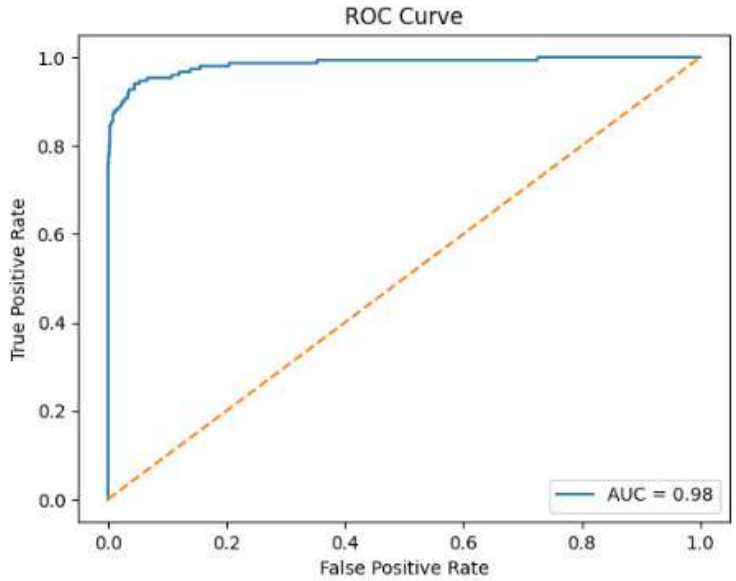
File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

```
plt.show()
```



🧪 Simulation Interface (Widgets Enabled)

[25]:

```
text_input = widgets.Textarea(
    value='',
    placeholder='Type email message here...',
    description='Message:',
    layout=widgets.Layout(width='750px', height='120px')
)

check_button = widgets.Button(
```

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simula

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

🧪 Simulation Interface (Widgets Enabled)

[25]:

```
text_input = widgets.Textarea(
    value='',
    placeholder='Type email message here...',
    description='Message:',
    layout=widgets.Layout(width='750px', height='120px')
)

check_button = widgets.Button(
    description='Check Spam',
    button_style='success'
)

output = widgets.Output()

def on_button_clicked(b):
    with output:
        clear_output()
        msg = text_input.value

        if msg.strip() == "":
            print("⚠️ Please enter a message")
            return

        clean = preprocess(msg)
        vec = tfidf.transform([clean]).toarray()
        pred = model.predict(vec)

        if pred[0] == 1:
            print("🚨 Result: SPAM MESSAGE")
        else:
            print("✅ Result: NOT SPAM MESSAGE")

check_button.on_click(on_button_clicked)

display(text_input, check_button, output)
```

Home | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Simulation_Widge | Spam_Email_Detection_Simulat

localhost:8888/notebooks/Spam_Email_Simulation_Widgets_Enabled.ipynb

Chat

jupyter Spam_Email_Simulation_Widgets_Enabled Last Checkpoint: 1 minute ago

File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

```
    button_style='success'
)

output = widgets.Output()

def on_button_clicked(b):
    with output:
        clear_output()
        msg = text_input.value

        if msg.strip() == "":
            print("⚠️ Please enter a message")
            return

        clean = preprocess(msg)
        vec = tfidf.transform([clean]).toarray()
        pred = model.predict(vec)

        if pred[0] == 1:
            print("🚨 Result: SPAM MESSAGE")
        else:
            print("✅ Result: NOT SPAM MESSAGE")

check_button.on_click(on_button_clicked)

display(text_input, check_button, output)
```

Message: nn

Check Spam

✅ Result: NOT SPAM MESSAGE

[ ]: