In [1]:

```
import pandas as pd
```

In [2]:

```
data=pd.read_csv("spam.csv", encoding="latin-1")
```

In [3]:

```
data.head(5)
```

Out[3]:

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

In [4]:

```
data.columns
```

Out[4]:

```
Index(['class', 'message', 'Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], dtype='object')
```

In [5]:

```
data.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
```

In [6]:

```
data.head()
```

Out[6]:

	class	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...

In [7]:

```
data['class']=data['class'].map({'ham':0,'spam':1})
```

In [8]:

```
data.head()
```

Out[8]:

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

In [17]:

```
##NLP Technique
```

```
from sklearn.feature_extraction.text import CountVectorizer
```

In [18]:
```
cv=CountVectorizer()
```

In [19]:
```
x=data['message']
y=data['class']
```

In [20]:
```
x.shape
```

Out[20]:
```
(5572,)
```

In [21]:
```
y.shape
```

Out[21]:
```
(5572,)
```

In [22]:
```
x=cv.fit_transform(x)
```

In [23]:
```
x
```

Out[23]:
```
<5572x8672 sparse matrix of type '<class 'numpy.int64'>'
        with 73916 stored elements in Compressed Sparse Row format>
```

In [24]:
```
from sklearn.model_selection import train_test_split
```

In [25]:
```
x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.2)
```

In [26]:
```
x_train.shape
```

Out[26]:
```
(4457, 8672)
```

In [27]:
```
from sklearn.naive_bayes import MultinomialNB
```

In [28]:
```
model=MultinomialNB()
```

In [29]:
```
model.fit(x_train, y_train)
```

Out[29]:
```
MultinomialNB()
```

In [31]:
```
result=model.score(x_test, y_test)
```

In [32]:
```
result=result*100
```

In [33]:
```
result
```

Out[33]:
```
97.75784753363229
```

In [34]:
```
import pickle
```

In [35]:
```
pickle.dump(model, open("spam.pkl","wb"))
```

In [36]:
```
pickle.dump(cv, open("vectorizer.pkl","wb"))
```

In [37]:

```
clf=pickle.load(open("spam.pkl","rb"))
```

In [38]:

```
clf
```

Out[38]:

```
MultinomialNB()
```

In [40]:

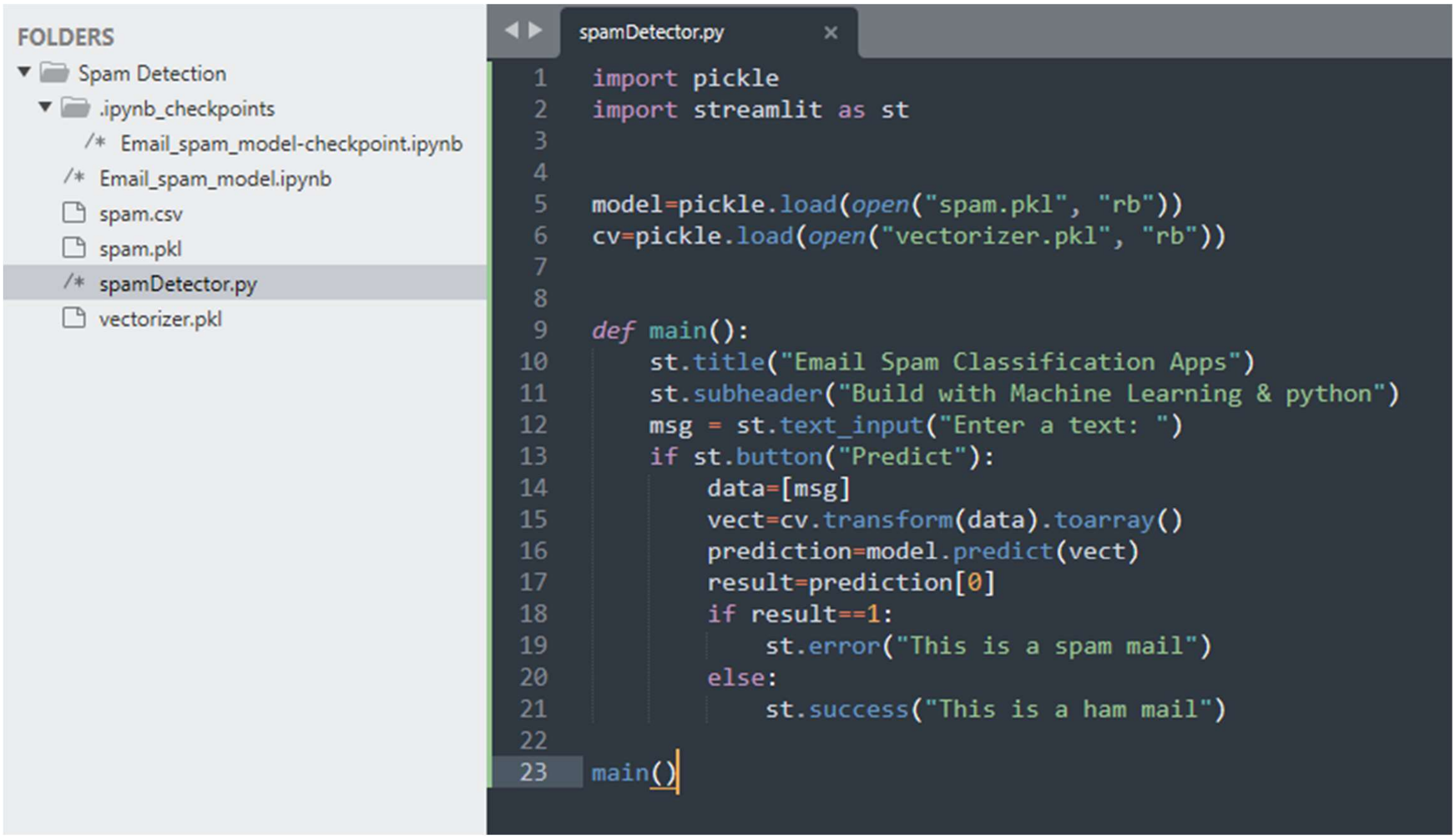
```
msg="hello there"
data=[msg]
vect=cv.transform(data).toarray()
result=model.predict(vect)
print(result)
```

```
[0]
```

In [ ]:

Sublime text editor:

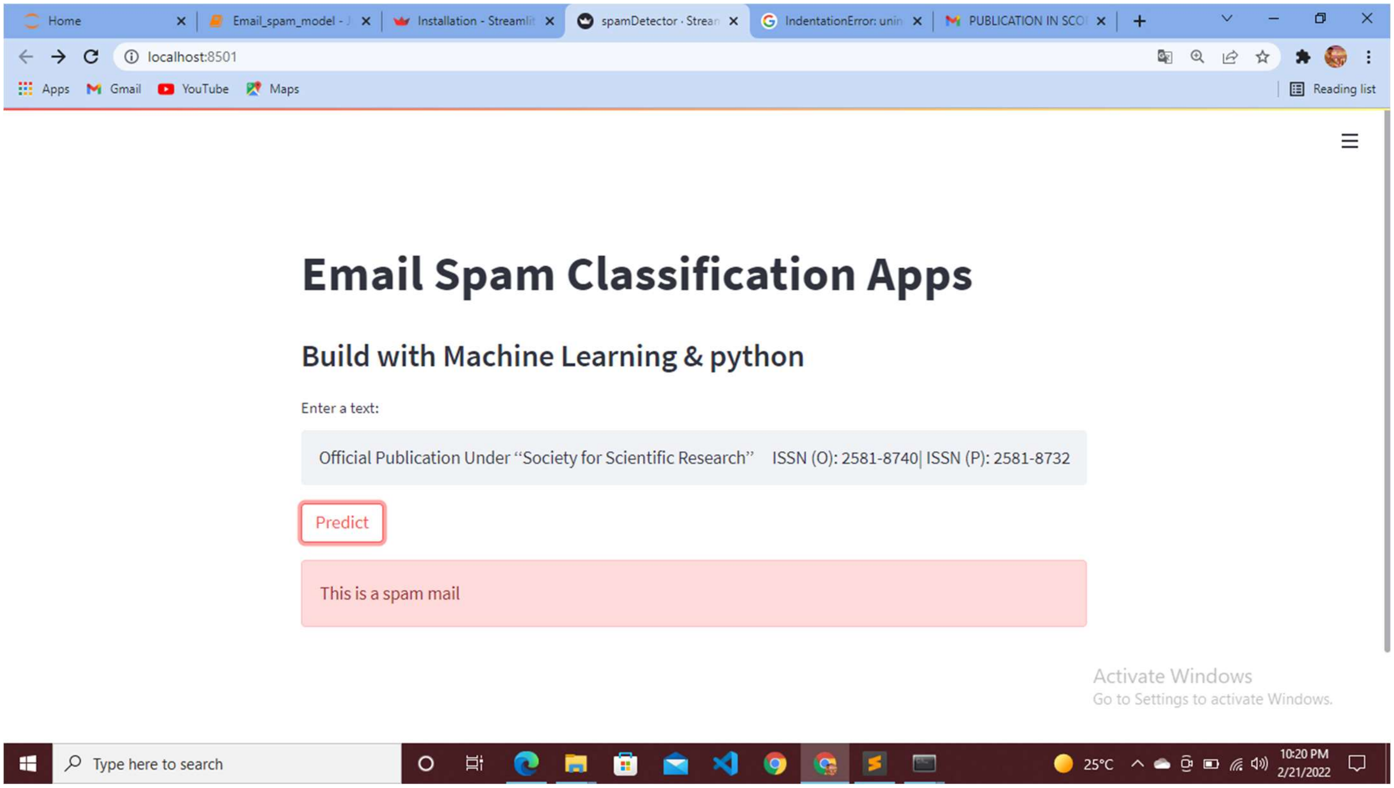
We use sublime text as editor.



```python
import pickle
import streamlit as st


model=pickle.load(open("spam.pkl", "rb"))
cv=pickle.load(open("vectorizer.pkl", "rb"))


def main():
    st.title("Email Spam Classification Apps")
    st.subheader("Build with Machine Learning & python")
    msg = st.text_input("Enter a text: ")
    if st.button("Predict"):
        data=[msg]
        vect=cv.transform(data).toarray()
        prediction=model.predict(vect)
        result=prediction[0]
        if result==1:
            st.error("This is a spam mail")
        else:
            st.success("This is a ham mail")

main()
```

User Interface:

We use Streamlit as python framework for our user interface. For user inputs we use random mail from our inbox and spam folder to detect spam or ham.



Email Spam Classification Apps

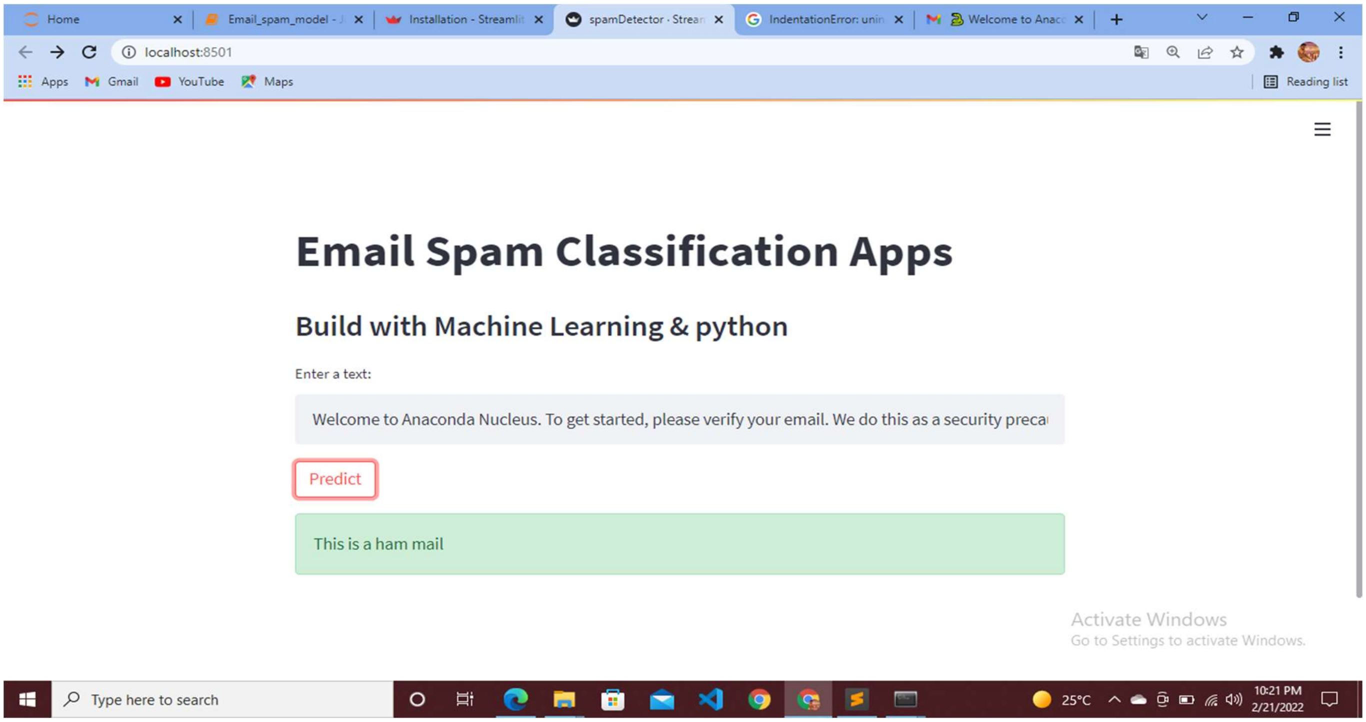
Build with Machine Learning & python

Enter a text:

Official Publication Under "Society for Scientific Research" ISSN (O): 2581-8740| ISSN (P): 2581-8732

Predict

This is a spam mail



Email Spam Classification Apps

Build with Machine Learning & python

Enter a text:

Welcome to Anaconda Nucleus. To get started, please verify your email. We do this as a security precai

Predict

This is a ham mail