**SMS Spam Classification**

# Download The Dataset Import The Required Library

*#importing the reqired Library*

import numpy as np import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns

import tensorflow as tf from tensorflow import keras

from tensorflow.keras import layers

# Read the Dataset

*# Reading the data*

df = pd.read\_csv("/content/spam.csv",encoding='latin-1') df.head()

v1 v2 Unnamed: 2

\

0 ham Go until jurong point, crazy.. Available only ... NaN

1 ham Ok lar... Joking wif u oni... NaN

2 spam Free entry in 2 a wkly comp to win FA Cup fina... NaN

3 ham U dun say so early hor... U c already then say... NaN

4 ham Nah I don't think he goes to usf, he lives aro... NaN

Unnamed: 3 Unnamed: 4

|  |  |  |
| --- | --- | --- |
| 0 | NaN | NaN |
| 1 | NaN | NaN |
| 2 | NaN | NaN |
| 3 | NaN | NaN |
| 4 | NaN | NaN |

df = df.drop(['Unnamed: 2','Unnamed: 3','Unnamed: 4'],axis=1)

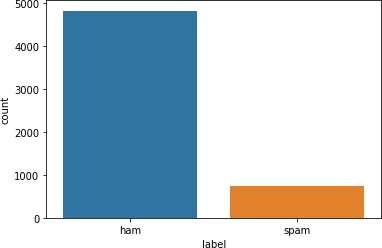
df = df.rename(columns={'v1':'label','v2':'Text'})

df['label\_enc'] = df['label'].map({'ham':0,'spam':1}) df.head()

label Text label\_enc

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

sns.countplot(x=df['label']) plt.show()



*# Find average number of tokens in all sentences* avg\_words\_len=round(sum([len(i.split()) **for** i **in** df['Text']])/len(df['Text'])) print(avg\_words\_len)

15

*# Splitting data for Training and testing*

from sklearn.model\_selection import train\_test\_split

X, y = np.asanyarray(df['Text']), np.asanyarray(df['label\_enc']) new\_df = pd.DataFrame({'Text': X, 'label': y})

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

new\_df['Text'], new\_df['label'], test\_size=0.2, random\_state=42)

X\_train.shape, y\_train.shape, X\_test.shape, y\_test.shape ((4457,), (4457,), (1115,), (1115,))

**def** word\_count\_plot(data):

*# finding words along with count*

word\_counter = collections.Counter([word **for** sentence **in** data **for**

word **in** sentence.split()])

most\_count = word\_counter.most\_common(30) *# 30 most common words # sorted data frame*

most\_count = pd.DataFrame(most\_count, columns=["Word", "Count"]).sort\_values(by="Count")

most\_count.plot.barh(x = "Word", y = "Count", color="green", figsize=(10, 15))

# Create a Model

from sklearn.feature\_extraction.text import TfidfVectorizer from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import classification\_report,accuracy\_score

tfidf\_vec = TfidfVectorizer().fit(X\_train) X\_train\_vec,X\_test\_vec = tfidf\_vec.transform(X\_train),tfidf\_vec.transform(X\_test)

baseline\_model = MultinomialNB() baseline\_model.fit(X\_train\_vec,y\_train)

MultinomialNB()

ham\_words = '' spam\_words = ''

*#creating an embedding layer*

*# load the whole embedding into memory*

embeddings\_index = dict()

f = open("/content/spam.csv")

**for** line **in** f:

values = line.split() word = values[0]

coefs = np.asarray(values[1:], dtype='float32') embeddings\_index[word] = coefs

f.close()

print('Loaded %s word vectors.' % len(embeddings\_index))

----------------------------------------------------------------------

-----

UnicodeDecodeError Traceback (most recent call last)

<ipython-input-8-ad0b3449a723> in <module>

* 1. embeddings\_index = dict()
  2. f = open("/content/spam.csv")

----> 6 for line in f:

1. values = line.split()
2. word = values[0]

/usr/lib/python3.7/codecs.py in decode(self, input, final)

1. # decode input (taking the buffer into account)
2. data = self.buffer + input

--> 322 (result, consumed) = self.\_buffer\_decode(data, self.errors, final)

1. # keep undecoded input until the next call
2. self.buffer = data[consumed:]

UnicodeDecodeError: 'utf-8' codec can't decode bytes in position 606- 607: invalid continuation byte

import pandas as pd import numpy as np import re

import collections import seaborn as sns

import matplotlib.pyplot as plt plt.style.use('dark\_background') import nltk

from nltk.stem import WordNetLemmatizer from nltk.corpus import stopwords import warnings

warnings.simplefilter(action='ignore', category=Warning) import keras

from keras.layers import Dense, Embedding, LSTM, Dropout from keras.models import Sequential

from keras.preprocessing.text import Tokenizer import pickle

**for** val **in** data[data['label'] == 'spam'].text: text = val.lower()

tokens = nltk.word\_tokenize(text)

**for** words **in** tokens:

spam\_words = spam\_words + words + ' '

----------------------------------------------------------------------

-----

NameError Traceback (most recent call last)

<ipython-input-6-ed68ec7f9b51> in <module>

----> 1 for val in data[data['label'] == 'spam'].text:

* 1. text = val.lower()
  2. tokens = nltk.word\_tokenize(text)
  3. for words in tokens:
  4. spam\_words = spam\_words + words + ' '

NameError: name 'data' is not defined

from sklearn.preprocessing import LabelEncoder

lb\_enc = LabelEncoder()

y = lb\_enc.fit\_transform(data["SpamHam"])

tokenizer = Tokenizer() *#initializing the tokenizer* tokenizer.fit\_on\_texts(X)

*# fitting on the sms data*

text\_to\_sequence = tokenizer.texts\_to\_sequences(X)

# Fit the model

*#fit the model* history=model.fit(sequences\_matrix,Y\_train,batch\_size=20,epochs=15, validation\_split=0.2)

# Save the model

*#save the model*

model.save('A4Spam\_sms\_classifier.h5')

# Compile the Model

*#compile the model*

model.compile(loss='binary\_crossentropy',optimizer=Adam(),metrics=['ac curacy'])

# Test the model

test\_sequences = tok.texts\_to\_sequences(X\_test) test\_sequences\_matrix = keras.utils.pad\_sequences(test\_sequences,maxlen=max\_len) accr = model.evaluate(test\_sequences\_matrix,Y\_test) print('Test set\n Loss: {:0.3f}\n Accuracy:

{:0.3f}'.format(accr[0],accr[1]))