```
# Import all the Dependencies

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.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Data Collection & Preprocessing
# Load csv file into a pandas dataframe
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>

Saving emails.csv to emails.csv
```

```
# Load csv file into a pandas dataframe
raw_mail_data = pd.read_csv('/content/emails.csv')

# Print the dataset
print(raw_mail_data)
```

```
                                                   text  spam
0     Subject: naturally irresistible your corporate...     1
1     Subject: the stock trading gunslinger  fanny i...     1
2     Subject: unbelievable new homes made easy  im ...     1
3     Subject: 4 color printing special  request add...     1
4     Subject: do not have money , get software cds ...     1
...                                                 ...   ...
5723  Subject: re : research and development charges...     0
5724  Subject: re : receipts from visit  jim ,  than...     0
5725  Subject: re : enron case study update  wow ! a...     0
5726  Subject: re : interest  david ,  please , call...     0
5727  Subject: news : aurora 5 . 2 update  aurora ve...     0

[5728 rows x 2 columns]
```

```
# replace the null values with a null string
mail_data = raw_mail_data.where((pd.notnull(raw_mail_data)),'')

# Print the first five rows of the dataframe
mail_data.head()
```


```
{"summary":"{\n  \"name\": \"mail_data\",\n  \"rows\": 5728,\n
\"fields\": [\n    {\n      \"column\": \"text\",\n
\"properties\": {\n        \"dtype\": \"string\",\n
\"num_unique_values\": 5695,\n        \"samples\": [\n
\"Subject: eprm article  hi vince ,  ?  as always , it was good to see
you again in houston - we all enjoyed the meal  very much , the
```

restaurant was a good choice . ? it ' s that time again i ' m afraid . can you pls cast your eye over the attached ? and , if at all possible , get back to me in the next few days - i have to deliver something to london by friday . ? how ' s the course going at rice ? not too much work i hope . ? best regards . ? chris . ? - eprm _ 09 _ fwd _ vol _ estimation . doc\",\n \"Subject: fluid analysis our customer speak volumes about our spur m product \\\\\" i just wanted to write and thank you for spur - m . i suffered from poor sperm count and motility . i found your site and ordered spur - m fertility blend for men . i have wondered for years what caused low semen and sperm count , and how i could improve my fertility and help my wife conceive . spur - m seems to have done just that ! thank you for your support . \\\\\" andrew h . , london , uk \\\\\" spur - m really does help improve fertility and effectiveness of sperm and semen motility . i used it for the past few months , and not only does it work - i also feel better to . i have more energy . this is an excellent counter to low sperm count and motility . i ' ll be buying more ! ! ! \\\\\" franz k . , bonn , germany http : / / findgoodstuffhere . com / spur / for removing , pls go here http : / / findgoodstuffhere . com / rm . php\",\n \"Subject: re : liquids limits oct . 20 john : i will be here most of the week , and am looking forward to working with niamh c . i will also check the availability of people in vince k . group as well as naveen andrews in ours . regards bjorn h . john l nowlan 24 / 10 / 2000 10 : 32 to : bjorn hagelmann / hou / ect @ ect cc : ted murphy / hou / ect @ ect subject : re : liquids limits oct . 20 bjorn , niamh clarke is going to come to houston from mon afternoon to friday next week to work on nvar . she developed var models for mitsubishi and has lots of experience in this area . can you please provide her with the best people we can from research and rac so we can try and get a better understanding and more confidence in our model . i ' m sure you agree with me that if my group is going to make any progress we need to get this sorted . thanks in advance . - forwarded by john l nowlan / hou / ect on 10 / 24 / 2000 09 : 51 am - from : bjorn hagelmann 10 / 24 / 2000 07 : 31 am to : john l nowlan / hou / ect @ ect cc : scott earnest / hou / ect @ ect subject : re : liquids limits oct . 20 i think we need to sit down and talk about developing reporting that will show the risk in the books . at this point and time it can be derived , but only if you know what to look for . i would appreciate if you had some time to do so . regards bjorn h john l nowlan 23 / 10 / 2000 13 : 10 to : christian lebroc / corp / enron @ enron , scott earnest / hou / ect @ ect , bjorn hagelmann / hou / ect @ ect cc : subject : re : liquids limits oct . 20 looking at these numbers i think the var model must be waaaaaaaaaay over calcing something , most likely the spreads . the net and outright product position are negligible . seems it would take one hell of a daily move to loose 12 . 7 on these positions .\"\n],\n \"semantic_type\": \"\",\n

\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"spam\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n
\"num_unique_values\": 2,\n   \"samples\": [\n   0,\n
1\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"mail_data"}

```
# Checking the number of rows & columns
mail_data.shape
```

```
(5728, 2)
```

```
# how many are spam and ham
mail_data.spam.value_counts()
```

```
spam
0    4360
1    1368
Name: count, dtype: int64
```

```
# Split the data into feautures & targets
X = mail_data['text']
Y = mail_data['spam']

print(Y)
```

```
0       1
1       1
2       1
3       1
4       1
       ..
5723    0
5724    0
5725    0
5726    0
5727    0
Name: spam, Length: 5728, dtype: int64
```

```
print(X)
```

```
0       Subject: naturally irresistible your corporate...
1       Subject: the stock trading gunslinger  fanny i...
2       Subject: unbelievable new homes made easy  im ...
3       Subject: 4 color printing special  request add...
4       Subject: do not have money , get software cds ...
                              ...
5723    Subject: re : research and development charges...
5724    Subject: re : receipts from visit  jim ,  than...
5725    Subject: re : enron case study update  wow ! a...
```

```
5726    Subject: re : interest  david ,  please , call...
5727    Subject: news : aurora 5 . 2 update  aurora ve...
Name: text, Length: 5728, dtype: object
```

```
# Splitting the data into training data and test data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.2, random_state=3)

print(X.shape)
print(X_train.shape)
print(X_test.shape)
```

```
(5728,)
(4582,)
(1146,)
```

```
# Convert text data into meaningful numerical values
# Feature extraction
# Transform text data into feature vectors that can be used in our
logistic regression model
# TfidfVectorizer - if a word repeated several ties its given a score.
if a word appears miniscule times its given a score.

feature_extraction = TfidfVectorizer(min_df=1, stop_words='english',
lowercase=True)

X_train_features = feature_extraction.fit_transform(X_train)
X_test_features = feature_extraction.transform(X_test)

# Convert Y train and Y test as intergers
Y_train = Y_train.astype('int')
Y_test = Y_test.astype('int')

# Print X test & X train
print(X_train_features)
```

```
  (0, 29045)    0.027350831183146494
  (0, 12734)    0.2731330732901836
  (0, 9702)     0.2434779314354512
  (0, 26414)    0.20469906162468185
  (0, 11945)    0.16768019627120936
  (0, 21847)    0.3221661388560145
  (0, 29259)    0.19594745660827412
  (0, 7447)     0.26830087740346925
  (0, 8787)     0.13947216511966962
  (0, 17361)    0.14810582386362775
  (0, 8653)     0.07145974179954041
  (0, 7094)     0.07086072273068604
  (0, 33101)    0.060807769107754726
  (0, 8799)     0.13947216511966962
  (0, 25415)    0.04547507625063386
```

```
  (0, 14190)    0.20469906162468185
  (0, 14051)    0.18776782473981463
  (0, 30580)    0.1357750225763003
  (0, 24306)    0.048890118949862775
  (0, 4932)     0.10871375473306225
  (0, 14192)    0.11948158504424808
  (0, 15229)    0.07973851459430858
  (0, 27608)    0.09497852966241772
  (0, 19873)    0.04042751503749571
  (0, 30155)    0.04986444772716652
  :   :
  (4581, 29985) 0.05731960277359003
  (4581, 2039)  0.11904749726535191
  (4581, 17129) 0.11305487382993415
  (4581, 112)   0.08190880155517595
  (4581, 1267)  0.10098266909611253
  (4581, 16993) 0.1031014024665001
  (4581, 4987)  0.11815520710746141
  (4581, 32353) 0.10192506274469548
  (4581, 18624) 0.06222322140391686
  (4581, 429)   0.07532196662652385
  (4581, 3279)  0.12232426528266004
  (4581, 27437) 0.08739417136034547
  (4581, 14328) 0.11501159908852511
  (4581, 10327) 0.09663204231271408
  (4581, 5261)  0.08190880155517595
  (4581, 22004) 0.14873331081277677
  (4581, 19871) 0.09663204231271408
  (4581, 25900) 0.12401650772265771
  (4581, 20409) 0.4174639199214749
  (4581, 1598)  0.13490546102102918
  (4581, 19031) 0.13839809802353042
  (4581, 4798)  0.5922167455482086
  (4581, 14760) 0.21002774702854982
  (4581, 20083) 0.21002774702854982
  (4581, 15485) 0.15224117631765458
```

```
# Training the Logistic Regression model
model = LogisticRegression()

# Training Logistic Regression model with training data
model.fit(X_train_features, Y_train)
```

```
LogisticRegression()
```

```
# Model evaluation of the trained data
prediction_on_training_data = model.predict(X_train_features)
accuracy_on_training_data = accuracy_score(Y_train,
prediction_on_training_data)
```

```
print('accuracy_on_training_data:', accuracy_on_training_data)
```

```
accuracy_on_training_data: 0.9958533391532082
```

```
# Accuracy score on training data = 99.5%
# Model performed well.

# Model Evaluation on test data
prediction_on_test_data = model.predict(X_test_features)
accuracy_on_test_data = accuracy_score(Y_test,
prediction_on_test_data)

print('accuracy_on_test_data:' , accuracy_on_test_data)
```

```
accuracy_on_test_data: 0.9834205933682374
```

```
# Accuracy score on test data is 98.3$
# Model performed well.

# Building a predictive system
input_mail = [" naturally irresistible your corporate identity  lt is
really hard to recollect a company : the  market is full of
suqgestions and the information isoverwhelminq ; but a good  catchy
logo , stylish statlonery and outstanding website  will make the task
much easier .  we do not promise that havinq ordered a iogo your
company will automaticaily become a world ieader : it isguite ciear
that  without good products , effective business organization and
practicable aim it  will be hotat nowadays market ; but we do promise
that your marketing efforts  will become much more effective . here is
the list of clear  benefits : creativeness : hand - made , original
logos , specially done  to reflect your distinctive company image .
convenience : logo and stationery  are provided in all formats ; easy
- to - use content management system letsyou  change your website
content and even its structure . promptness : you  will see logo
drafts within three business days . affordability : your  marketing
break - through shouldn ' t make gaps in your budget . 100 %
satisfaction  guaranteed "]

# Convert text to feature vectors

input_data_features = feature_extraction.transform(input_mail)

# Makung predictions

prediction = model.predict(input_data_features)

# Print the predicted value
print(prediction)

if prediction[0]==1:
  print('Ham mail')
```

```
else:
  print('Spam mail')

[1]
Ham mail
```