In [1]:
```
!pip freeze > kaggle_image_requirements.txt
```

Lea y pre-procese el dataset Enron

Lea el dataset Enron y obtenga una idea de los datos imprimiendo mensajes de muestra en la pantalla

In [14]:
```
import numpy as np # linear algebra
import pandas as pd # procesando datos, CSV file I/O (e.g. pd.read_csv)

# Input data files disponibles en el directorio "../input/".
filepath = "/Users/marcelo/Documents/emails.csv"

# Leer los datos enron en un pandas.DataFrame llamada emails
emails = pd.read_csv(filepath)

print("¡Se cargaron exitosamente {} filas and {} columnas!".format(emails.sh
print(emails.head())
```

```
¡Se cargaron exitosamente 517401 filas and 2 columnas!
                      file                                                messa
ge
0     allen-p/_sent_mail/1.  Message-ID: <18782981.1075855378110.JavaMail.
e...
1    allen-p/_sent_mail/10.  Message-ID: <15464986.1075855378456.JavaMail.
e...
2   allen-p/_sent_mail/100.  Message-ID: <24216240.1075855687451.JavaMail.
e...
3  allen-p/_sent_mail/1000.  Message-ID: <13505866.1075863688222.JavaMail.
e...
4  allen-p/_sent_mail/1001.  Message-ID: <30922949.1075863688243.JavaMail.
e...
```

In [25]:
```
# 1) DESPLIEGUE CON MAYOR DETALLE EL SEGUNDO EMAIL
emails.iloc[1]
```

Out[25]:
```
file                                  allen-p/_sent_mail/10.
message    Message-ID: <15464986.1075855378456.JavaMail.e...
Name: 1, dtype: object
```

Separar los encabezados (headers) de los cuerpos del mensaje

In [30]:
```
import email

def extract_messages(df):
    messages = []
    for item in df["message"]:
        # Regresar una estructura de objeto de mensaje a partir de un string
        e = email.message_from_string(item)
        # obtener el cuerpo del mensaje
        message_body = e.get_payload()
```

```
        messages.append(message_body)
    print("¡Se recuperó existosamente el cuerpo del mensaje a partir de los
    return messages

#2) EXTRAIGA LOS MENSAJES DE LOS EMAILS EN UNA LISTA LLAMADA bodies, usando

bodies = extract_messages(emails)
```

```
¡Se recuperó existosamente el cuerpo del mensaje a partir de los e-mails!
```

In [37]:
```
##3) GENERE UN DATAFRAME LLAMADO bodies_df UTILIZANDO LA BIBLIOTECA PANDAS,
##USE LA FUNCIÓN RANDOM.SAMPLE PARA OBTENER LA MUESTRA. ESTO PERMITE CREAR N
import random
bodies_df = pd.DataFrame(random.sample(bodies, 10000), columns=['messages'])
#expandir las opciones de display default de pandas para hacer los emails má
pd.set_option('display.max_colwidth', 300)

bodies_df.head() # podrían hacer print(bodies_df.head()), pero Jupyter despl
```

Out[37]:

	messages
0	\n\n-----Original Message-----\nFrom: Braddock, Billy \nSent: Monday, November 26, 2001 11:15 AM\nTo: Kroll, Heather\nSubject: A luxurious resort - La Casa Que Canta\n\n\n \n \n http://www.lacasaquecanta.com/
1	fyi\n---------- Forwarded by Stephanie Miller/Corp/Enron on 04/06/2001 \n11:26 AM ----------\n\n\nLinda Roberts\n04/06/2001 09:53 AM\nTo: dianan@calpine.com, bdaugherty@aep.com, pookies@ev1.net, oakley@pdq.net, \nlstephenson@reliantenergy.com, mstephenson@pcenergy.co...
2	Not much back from Tony, but he did find a Chicago + .06 offer for norther \nborder gas. He will keep looking. Let me know if you need anything in the \nmeantime.\n\nRichard\n\n
3	\n---------- Forwarded by Mark Dana Davis/HOU/ECT on 09/17/2001 07:04 AM ----------\n\n\n"Mark Davis" <mddjunior31@earthlink.net> on 09/16/2001 01:27:06 AM\nPlease respond to mddjunior31@earthlink.net\nTo:\tmdavis@enron.com, "Superjohn5" <Superjohn5@hotmail.com>\ncc:...
4	This book request has been completed. Please call me when you have any questions.\n\nThank you.\n\nJennifer Velasco\nRisk Controls\nX 5-2526\n\n

El siguiente código (comentado) es posiblemente la forma más al estilo Python de lograr la extracción de los cuerpos de los mensajes. Tiene solo 2 líneas y logra el mismo resultado. Sin embargo, creemos que el código anterior es más transparente con respecto a cómo se lleva a cabo el procesamiento y, como tal, lo dejamos aquí para los expertos en Python si así lo prefieren.

In [6]:
```
#messages = emails["message"].apply(email.message_from_string)
#bodies_df = messages.apply(lambda x: x.get_payload()).sample(10000)
```

Lea y Preprocese el Corpus de Email Fraudulent "419"

In [39]:
```
filepath = "/Users/marcelo/Documents/fradulent_emails.txt"
with open(filepath, 'r',encoding="latin1") as file:
    data = file.read()
```

Imprima los primeros 20000 caracteres de read file string (esto nos da unos pocos emails), y dése cuenta del keyword `From r` cercano al inicio de cada encabezado de email

In [40]:
```
print(data[:20000])
```

From r Wed Oct 30 21:41:56 2002
Return-Path: <james_ngola2002@maktoob.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <james_ngola2002@maktoob.com>
Message-Id: <200210310241.g9V2fNm6028281@cs.CU>
From: "MR. JAMES NGOLA." <james_ngola2002@maktoob.com>
Reply-To: james_ngola2002@maktoob.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 02:38:20 +0000
Subject: URGENT BUSINESS ASSISTANCE AND PARTNERSHIP
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id g9V2foW24311
Status: O

FROM:MR. JAMES NGOLA.
CONFIDENTIAL TEL: 233-27-587908.
E-MAIL: (james_ngola2002@maktoob.com).

URGENT BUSINESS ASSISTANCE AND PARTNERSHIP.

DEAR FRIEND,

I AM (DR.) JAMES NGOLA, THE PERSONAL ASSISTANCE TO THE LATE CONGOLESE (PRES IDENT LAURENT KABILA) WHO WAS ASSASSINATED BY HIS BODY GUARD ON 16TH JAN. 20 01.

THE INCIDENT OCCURRED IN OUR PRESENCE WHILE WE WERE HOLDING MEETING WITH HIS EXCELLENCY OVER THE FINANCIAL RETURNS FROM THE DIAMOND SALES IN THE AREAS CO NTROLLED BY (D.R.C.) DEMOCRATIC REPUBLIC OF CONGO FORCES AND THEIR FOREIGN A LLIES ANGOLA AND ZIMBABWE, HAVING RECEIVED THE PREVIOUS DAY (USD$100M) ONE H UNDRED MILLION UNITED STATES DOLLARS, CASH IN THREE DIPLOMATIC BOXES ROUTED THROUGH ZIMBABWE.

MY PURPOSE OF WRITING YOU THIS LETTER IS TO SOLICIT FOR YOUR ASSISTANCE AS T O BE A COVER TO THE FUND AND ALSO COLLABORATION IN MOVING THE SAID FUND INTO YOUR BANK ACCOUNT THE SUM OF (USD$25M) TWENTY FIVE MILLION UNITED STATES DOL LARS ONLY, WHICH I DEPOSITED WITH A SECURITY COMPANY IN GHANA, IN A DIPLOMAT IC BOX AS GOLDS WORTH (USD$25M) TWENTY FIVE MILLION UNITED STATES DOLLARS ON LY FOR SAFE KEEPING IN A SECURITY VAULT FOR ANY FURTHER INVESTMENT PERHAPS I N YOUR COUNTRY.

YOU WERE INTRODUCED TO ME BY A RELIABLE FRIEND OF MINE WHO IS A TRAVELLER,AN D ALSO A MEMBER OF CHAMBER OF COMMERCE AS A RELIABLE AND TRUSTWORTHY PERSON WHOM I CAN RELY ON AS FOREIGN PARTNER, EVEN THOUGH THE NATURE OF THE TRANSAC TION WAS NOT REVEALED TO HIM FOR SECURITY REASONS.

THE (USD$25M) WAS PART OF A PROCEEDS FROM DIAMOND TRADE MEANT FOR THE LATE P RESIDENT LAURENT KABILA WHICH WAS DELIVERED THROUGH ZIMBABWE IN DIPLOMATIC B OXES. THE BOXES WERE KEPT UNDER MY CUSTODY BEFORE THE SAD EVENT THAT TOOK TH

E LIFE OF (MR. PRESIDENT).THE CONFUSION THAT ENSUED AFTER THE ASSASSINATION AND THE SPORADIC SHOOTING AMONG THE FACTIONS, I HAVE TO RUN AWAY FROM THE CO UNTRY FOR MY DEAR LIFE AS I AM NOT A SOLDIER BUT A CIVIL SERVANT I CROSSED R IVER CONGO TO OTHER SIDE OF CONGO LIBREVILLE FROM THERE I MOVED TO THE THIRD COUNTRY GHANA WHERE I AM PRESENTLY TAKING REFUGE.

AS A MATTER OF FACT, WHAT I URGENTLY NEEDED FROM YOU IS YOUR ASSISTANCE IN M OVING THIS MONEY INTO YOUR ACCOUNT IN YOUR COUNTRY FOR INVESTMENT WITHOUT RA ISING EYEBROW. FOR YOUR ASSISTANCE I WILL GIVE YOU 20% OF THE TOTAL SUM AS Y OUR OWN SHARE WHEN THE MONEY GETS TO YOUR ACCOUNT, WHILE 75% WILL BE FOR ME, OF WHICH WITH YOUR KIND ADVICE I HOPE TO INVEST IN PROFITABLE VENTURE IN YOU R COUNTRY IN OTHER TO SETTLE DOWN FOR MEANINGFUL LIFE, AS I AM TIRED OF LIVI NG IN A WAR ENVIRONMENT.

THE REMAINING 5% WILL BE USED TO OFFSET ANY COST INCURRED IN THE CAUSE OF MO VING THE MONEY TO YOUR ACCOUNT. IF THE PROPOSAL IS ACCEPTABLE TO YOU PLEASE CONTACT ME IMMEDIATELY THROUGH THE ABOVE TELEPHONE AND E-MAIL, TO ENABLE ME ARRANGE FACE TO FACE MEETING WITH YOU IN GHANA FOR THE CLEARANCE OF THE FUND S BEFORE TRANSFRING IT TO YOUR BANK ACCOUNT AS SEEING IS BELIEVING.

FINALLY, IT IS IMPORTANT ALSO THAT I LET YOU UNDERSTAND THAT THERE IS NO RIS K INVOLVED WHATSOEVER AS THE MONEY HAD NO RECORD IN KINSHASA FOR IT WAS MEAN T FOR THE PERSONAL USE OF (MR. PRESIDEND) BEFORE THE NEFARIOUS INCIDENT OCC URRED, AND ALSO I HAVE ALL THE NECESSARY DOCUMENTS AS REGARDS TO THE FUNDS I NCLUDING THE (CERTIFICATE OF DEPOSIT), AS I AM THE DEPOSITOR OF THE CONSIGNM ENT.

LOOKING FORWARD TO YOUR URGENT RESPONSE.

YOUR SINCERELY,

MR. JAMES NGOLA.

From r Thu Oct 31 08:11:39 2002
Return-Path: <bensul2004nng@spinfinder.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <bensul2004nng@spinfinder.com>
Message-Id: <200210311310.g9VDANt24674@bloodwork.mr.itd.UM>
From: "Mr. Ben Suleman" <bensul2004nng@spinfinder.com>
Date: Thu, 31 Oct 2002 05:10:00
To: R@M
Subject: URGENT ASSISTANCE /RELATIONSHIP (P)
MIME-Version: 1.0
Content-Type: text/plain;charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
Status: O

Dear Friend,

I am Mr. Ben Suleman a custom officer and work as Assistant controller of th e Customs and Excise department Of the Federal Ministry of Internal Affairs stationed at the Murtala Mohammed International Airport, Ikeja, Lagos-Nigeri a.

After the sudden death of the former Head of state of Nigeria General Sanni Abacha on June 8th 1998 his aides and immediate members of his family were a rrested while trying to escape from Nigeria in a Chartered jet to Saudi Arab ia with 6 trunk boxes Marked "Diplomatic Baggage". Acting on a tip-off as th ey attempted to board the Air Craft,my officials carried out a thorough sear ch on the air craft and discovered that the 6 trunk boxes contained foreign currencies amounting to US$197,570,000.00(One Hundred and Ninety-Seven Mill ion Five Hundred Seventy Thousand United States Dollars).

I declared only (5) five boxes to the government and withheld one (1) in my custody containing the sum of (US$30,000,000.00) Thirty Million United State s Dollars Only, which has been disguised to prevent their being discovered d uring transportation process.Due to several media reports on the late head o f state about all the money him and his co-government officials stole from o ur government treasury amounting
to US$55 Billion Dollars (ref:ngrguardiannews.com) of July 2nd 1999. Even th e London times of July 1998 reported that General Abacha has over US$3.Billi on dollars in one account overseas. We decided to conceal this one (1)box ti ll the situation is calm and quite on the issue. The box was thus deposited with a security company here in Nigeria and tagged as "Precious Stones and J ewellry" in other that its
content will not be discovered. Now that all is calm, we (myself and two of my colleagues in the operations team) are now ready to move this box out of the country through a diplomatic arrangement which is the safest means.

However as government officials the Civil Service Code of Conduct does not a llow us by law to operate any foreign account or own foreign investment and the amount of money that can be found in our account
cannot be more than our salary on the average, thus our handicapp and our ne ed for your assistance to help collect and keep safely in your account this money.

Therefore we want you to assist us in moving this money out of Nigeria. We s hall definitely compensate you handsomely for the assistance. We can do this by instructing the Security Company here in Nigeria to
move the consignment to their affiliate branch office outside Nigeria throug h diplomatic means and the consignment will be termed as Precious Stones and Jewelleries" which you bought during your visit to Nigeria and is being tran sfered to your country from here for safe keeping. Then we can arrange to me et at the destination country to take the delivery of the consignment. You w ill thereafter open an account there and lodge the Money there and gradually instruct remittance to your Country.

This business is 100% risk free for you so please treat this matter with utm ost confidentiality .If you indicate your interest to assist us please just e-mail me for more Explanation on how we plan to execute the transaction.

Expecting your response urgently.

Best regards,

Mr. Ben Suleman

From r Thu Oct 31 17:27:16 2002
Return-Path: <obong_715@epatra.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <obong_715@epatra.com>
Message-Id: <200210312227.g9VMQvDj017948@bluewhale.cs.CU>
From: "PRINCE OBONG ELEME" <obong_715@epatra.com>
Reply-To: obong_715@epatra.com
To: webmaster@aclweb.org
Date: Thu, 31 Oct 2002 22:17:55 +0100
Subject: GOOD DAY TO YOU
X-Mailer: Microsoft Outlook Express 5.00.2919.6900DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id
g9VMRBW20642
Status: RO

FROM HIS ROYAL MAJESTY (HRM) CROWN RULER OF ELEME KINGDOM
CHIEF DANIEL ELEME, PHD, EZE 1 OF ELEME.E-MAIL
ADDRESS:obong_715@epatra.com

ATTENTION:PRESIDENT,CEO Sir/ Madam.

This letter might surprise you because we have met
neither in person nor by correspondence. But I believe
it is one day that you got to know somebody either in
physical or through correspondence.

I got your contact through discreet inquiry from the
chambers of commerce and industry of your country on
the net, you and your organization were revealed as
being quite astute in private entrepreneurship, one
has no doubt in your ability to handle a financialbusiness transaction.

However, I am the first son of His Royal
majesty,Obong.D. Eleme , and the traditional Ruler of
Eleme Province in the oil producing area of River
State of Nigeria. I am making this contact to you in
respect of US$60,000,000.00 (Sixty Million United
State Dollars), which I inherited, from my latefather.

This money was accumulated from royalties paid to my
father as compensation by the oil firms located in our
area as a result of oil presence on our land, which
hamper agriculture, which is our major source oflivelihood.

Unfortunately my father died from protracted
diabetes.But before his death he called my attention
and informed me that he lodged some funds on a two
boxes with a security firm with an open beneficiary

status. The lodgment security code number was also
revealed to me, he then advised me to look for a
reliable business partner abroad, that will assist me
in investing the money in a lucrative business as a
result of economic instability in Nigeria. So this is
the main reason why I am contacting you for us to move
this money from the security firm to any Country of
your choice for investment purpose.

So I will like you to be the ultimate beneficiary, so
that the funds can be moved in your name and
particulars to any Country of your choice where it
will be claimed and invested. Hence my father have had
intimated the security firm personnel that the
beneficiary of the box is his foreign partner whose
particulars will be forwarded to the firm when due.

But I will guide you Accordingly. As soon as the funds
reach, I will then come over to meet you in person, so
that we can discuss physically on investment
potentials. Based on this assistance my Family and I
have unanimously decided to give you 30% of the total
money, 5% for Charity home, 10% for expenses, which
may arise during this transaction, Fax and phone bills
inclusive. The balance of 55% you will invest andmanaged for my Family.

I hereby guarantee you that this is not government
money, it is not drug money and it is not money from
arms deal. Though you have to maintain high degree of
confidentiality on this matter. I will give more
details about the proceedings of this transaction as
soon as I receive your favorable reply.

Please reply to my Email Address:obong_715@epatra.com
I hope this will be the beginning of a prosperous
relationship between my family and your family.

Nevertheless if you are for any reason not interested,
kindly inform me immediately so that I will look foranother contact.

I am waiting for your quick response.

Yours faithfully,

Prince Obong Abbot

From r Thu Oct 31 17:53:56 2002
Return-Path: <obong_715@epatra.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <obong_715@epatra.com>
Message-Id: <200210312253.g9VMreDj018024@bluewhale.cs.CU>
From: "PRINCE OBONG ELEME" <obong_715@epatra.com>
Date: Thu, 31 Oct 2002 22:44:20
To: webmaster@aclweb.org
Subject: GOOD DAY TO YOU

MIME-Version: 1.0
Content-Type: text/plain;charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
Status: RO

FROM HIS ROYAL MAJESTY (HRM) CROWN RULER OF ELEME KINGDOM
CHIEF DANIEL ELEME, PHD, EZE 1 OF ELEME.E-MAIL
ADDRESS:obong_715@epatra.com

ATTENTION:PRESIDENT,CEO Sir/ Madam.

This letter might surprise you because we have met
neither in person nor by correspondence. But I believe
it is one day that you got to know somebody either in
physical or through correspondence.

I got your contact through discreet inquiry from the
chambers of commerce and industry of your country on
the net, you and your organization were revealed as
being quite astute in private entrepreneurship, one
has no doubt in your ability to handle a financialbusiness transaction.

However, I am the first son of His Royal
majesty,Obong.D. Eleme , and the traditional Ruler of
Eleme Province in the oil producing area of River
State of Nigeria. I am making this contact to you in
respect of US$60,000,000.00 (Sixty Million United
State Dollars), which I inherited, from my latefather.

This money was accumulated from royalties paid to my
father as compensation by the oil firms located in our
area as a result of oil presence on our land, which
hamper agriculture, which is our major source oflivelihood.

Unfortunately my father died from protracted
diabetes.But before his death he called my attention
and informed me that he lodged some funds on a two
boxes with a security firm with an open beneficiary
status. The lodgment security code number was also
revealed to me, he then advised me to look for a
reliable business partner abroad, that will assist me
in investing the money in a lucrative business as a
result of economic instability in Nigeria. So this is
the main reason why I am contacting you for us to move
this money from the security firm to any Country of
your choice for investment purpose.

So I will like you to be the ultimate beneficiary, so
that the funds can be moved in your name and
particulars to any Country of your choice where it
will be claimed and invested. Hence my father have had
intimated the security firm personnel that the
beneficiary of the box is his foreign partner whose
particulars will be forwarded to the firm when due.

But I will guide you Accordingly. As soon as the funds

reach, I will then come over to meet you in person, so
that we can discuss physically on investment
potentials. Based on this assistance my Family and I
have unanimously decided to give you 30% of the total
money, 5% for Charity home, 10% for expenses, which
may arise during this transaction, Fax and phone bills
inclusive. The balance of 55% you will invest andmanaged for my Family.

I hereby guarantee you that this is not government
money, it is not drug money and it is not money from
arms deal. Though you have to maintain high degree of
confidentiality on this matter. I will give more
details about the proceedings of this transaction as
soon as I receive your favorable reply.

Please reply to my Email Address:obong_715@epatra.com
I hope this will be the beginning of a prosperous
relationship between my family and your family.

Nevertheless if you are for any reason not interested,
kindly inform me immediately so that I will look foranother contact.

I am waiting for your quick response.

Yours faithfully,

Prince Obong Eleme

From r Fri Nov 1 04:48:39 2002
Return-Path: <m_abacha03@www.com>
X-Sieve: cmu-sieve 2.0
Return-Path: <m_abacha03@www.com>
Message-Id: <200211010948.gA19mLu22932@perfectworld.mr.itd.UM>
From: "Maryam Abacha" <m_abacha03@www.com>
Reply-To: m_abacha03@www.com
To: R@M
Date: Fri, 1 Nov 2002 01:45:04 +0100
Subject: I Need Your Assistance.
X-Mailer: Microsoft Outlook Express 5.00.2919.6900 DM
MIME-Version: 1.0
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: 8bit
X-MIME-Autoconverted: from quoted-printable to 8bit by sideshowmel.si.UM id
gA19mVW29040
Status: RO

Dear sir,

It is with a heart full of hope that I write to seek your help in respect of
the context below. I am Mrs. Maryam Abacha the former first lady of the form
er Military Head of State of Nigeria General Sani Abacha whose sudden death
occurred on 8th of June 1998 as a result of cardiac arrest (heart attack) wh
ile on the seat of power.
I have no doubt about your capability and good-will to assist me in receivin
g into your custody (for safety) the sum of US$25 Million willed and deposit
ed in my favour by my late husband in a security and deposit company. Though

my contact to you for this assistance is not anchored on any personal ecomme ndation, I pray your understanding, good will and sincere assistance to resp ond to this message with honest intensions and concern.
This money is currently deposited here with a security company as miscellane ous awaiting collection and according to the agreement entered into at the t ime of deposit between my late husband and the Security Company at the time of deposit; the collection centre is in Ghana. As it is legally required, th e administration of my late husband's properties is under the authority of t he family's lawyer Tony Musa.

My Dear JK, since the demise of my husband, the present regime has been prob ing my late husband's wealth and properties, the London Newsweek of 13th mar ch 2000 referred. The investigating team led by Enrico Monfrini, the lawyer acting on behalf of the Nigerian government has so far submitted their repor t and presently, some liquid cash and assets, movable and immovable, have be en frozen and seized both locally here and internationally and my last hope is rested on the immediate security of this fund in your custody. Also, John nie Cochran, the lawyer who defended OJ Simpson, has been brought in by the Abacha entourage to help them retain the disputed funds. Fortunately, our fa mily lawyer had secretly protected the "Personal will" of my husband from th e notice of the investigators and have strictly advised that the willed mone y be urgently moved into an overseas account of Trusted Foreign family frien d without delay. For security reasons and f!
urther advise,! no relations or fr!
iends of ours should be used; as this is a measure of security. The governme nt had earlier placed foreign travel embargo on all our family members and s eized all known local and International outfits of our business empire.
The situation has been so terrible that we are virtually living on the assis tance of well-wishers. In view of this plight, I expect you to be trustworth y and kind enough to respond to this "SOS" call to save my children and I fr om a hopeless future.
I hereby agree to compensate your sincere and candid effort, immensely, whic h will be discussed between you, the Attorney and I. On your immediate respo nse, the Attorney will travel to Ghana where both of you will meet, and ther eafter proceed to the deposit company for the claims.
The documentations include, most importantly, Power of Attorney and Certific ate of Deposit amongst few others. These documents will be sent to you by Fa x, as you would have to present them for proper claims.
Please, be rest assured that this transaction is completely safe and legal b ut must be kept strictly to yourself even after the funds have been secured into our custody. This is so because any leakage of information could ruin t he whole transaction.
Please all contacts must be made through my lawyer on his email: tony_m@lawy er.com This has to be so as he has been mandated to handle this matter and I have fully briefed him (my Attorney) on my contact to you. Due to my present circumstance, I have handed everything over to him to co - ordinate and fina lize with you and I will communicate you as at when necessary. Please for th e safety of this transaction, reply stating your phone and fax numbers to en able them contact you directly without running the risk of mail interceptio n.
I look forward to your quick response.
Best Wishes,
Mrs. Maryam Abacha

```
From r  Sat Nov  2 00:18:06 2002
Return-Path: <davidkuta@postmark.net>
X-Sieve: cmu-sieve 2.0
Return-Path: <davidkut
```

Dividir en la palabra clave `From r` que aparece cerca del comienzo de cada correo electrónico

In [41]:
```
fraud_emails = data.split("From r")

print("¡Cargó exitosamente {} emails de spam!".format(len(fraud_emails)))
```

```
¡Cargó exitosamente 3978 emails de spam!
```

In [42]:
```
fraud_bodies = extract_messages(pd.DataFrame(fraud_emails,columns=["message"
fraud_bodies_df = pd.DataFrame(fraud_bodies[1:])

fraud_bodies_df.head() # podrías hacer print(fraud_bodies_df.head()), pero J
```

```
¡Se recuperó existosamente el cuerpo del mensaje a partir de los e-mails!
```

Out[42]:

	0
0	FROM:MR. JAMES NGOLA.\nCONFIDENTIAL TEL: 233-27-587908.\nE-MAIL: (james_ngola2002@maktoob.com).\n\nURGENT BUSINESS ASSISTANCE AND PARTNERSHIP.\n\n\nDEAR FRIEND,\n\nI AM (DR.) JAMES NGOLA, THE PERSONAL ASSISTANCE TO THE LATE CONGOLESE (PRESIDENT LAURENT KABILA) WHO WAS ASSASSINATED BY HIS BODY G...
1	Dear Friend,\n\nI am Mr. Ben Suleman a custom officer and work as Assistant controller of the Customs and Excise department Of the Federal Ministry of Internal Affairs stationed at the Murtala Mohammed International Airport, Ikeja, Lagos-Nigeria.\n\nAfter the sudden death of the former Head of s...
2	FROM HIS ROYAL MAJESTY (HRM) CROWN RULER OF ELEME KINGDOM \nCHIEF DANIEL ELEME, PHD, EZE 1 OF ELEME.E-MAIL \nADDRESS:obong_715@epatra.com \n\nATTENTION:PRESIDENT,CEO Sir/ Madam. \n\nThis letter might surprise you because we have met\nneither in person nor by correspondence. But I believe\nit is...
3	FROM HIS ROYAL MAJESTY (HRM) CROWN RULER OF ELEME KINGDOM \nCHIEF DANIEL ELEME, PHD, EZE 1 OF ELEME.E-MAIL \nADDRESS:obong_715@epatra.com \n\nATTENTION:PRESIDENT,CEO Sir/ Madam. \n\nThis letter might surprise you because we have met\nneither in person nor by correspondence. But I believe\nit is...
4	Dear sir, \n \nIt is with a heart full of hope that I write to seek your help in respect of the context below. I am Mrs. Maryam Abacha the former first lady of the former Military Head of State of Nigeria General Sani Abacha whose sudden death occurred on 8th of June 1998 as a result of cardiac ...

Definir funciones de tokenización, eliminación de stop-words y eliminación de puntuación

Antes de continuar, debemos decidir cuántas muestras extraer de cada clase. También debemos decidir la cantidad máxima de tokens por correo electrónico y la longitud máxima de cada token. Esto se hace configurando los siguientes hiperparámetros generales

In [43]:
```
Nsamp = 1000 # número de muestras a generar en cada clase - 'spam', 'not spa
maxtokens = 200 # el número máximo de tokens por documento
maxtokenlen = 100 # la longitud máxima de cada token
```

Tokenization

In [44]:
```
def tokenize(row):
    if row is None or row is '':
        tokens = ""
    else:
        tokens = str(row).split(" ")[:maxtokens]
    return tokens
```

```
<>:2: SyntaxWarning: "is" with a literal. Did you mean "=="?
<>:2: SyntaxWarning: "is" with a literal. Did you mean "=="?
/var/folders/60/x1l1rxds79nbrmd88tmm_yvw0000gr/T/ipykernel_20729/3728645439.
py:2: SyntaxWarning: "is" with a literal. Did you mean "=="?
  if row is None or row is '':
```

Usar regular expressions para quitar caracteres no necesarios

Después, definimos una función para eliminar los signos de puntuación y otros caracteres no alfabéticos (usando expresiones regulares) de los correos electrónicos con la ayuda de la omnipresente biblioteca de expresiones regulares de Python. En el mismo paso, truncamos todos los tokens a la hiperparámetro `maxtokenlen` definido anteriormente

In [46]:
```
import re

def reg_expressions(row):
    tokens = []
    try:
        for token in row:
            #4) ESCRIBA UNA LÍNEA DE CÓDIGO QUE CONVIERTA UNA VARIABLE LLAMA
            token = token.lower()
            token = re.sub(r'[\W\d]', "", token)
            #5) TRUNQUE EL TOKEN A LA MÁXIMA LONGITUD ANTES DEFINIDA
            token = token[:maxtokenlen]
            tokens.append(token)
    except:
        token = ""
        tokens.append(token)
    return tokens
```

Eliminación de Stop-word

Definamos una función para eliminar las stopwords —palabras que ocurren con tanta frecuencia en el lenguaje que no ofrecen información útil para la clasificación. Esto incluye palabras como “the” y “are”, y la popular biblioteca NLTK proporciona una lista ampliamente utilizada que se empleará.

In [50]:
```
import nltk

nltk.download('stopwords')
from nltk.corpus import stopwords
stopwords = stopwords.words('english')

def stop_word_removal(row):
    ###6) CREE UNA NUEVA LISTA LLAMADA TOKEN, QUE CONTENGA SOLO AQUELLOS TOK
    token = [word for word in row if word not in stopwords]
    token = filter(None, token)
    return token
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/marcelo/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Modelo Bag-of-words

Para que la computadora pueda hacer inferencias sobre los correos electrónicos, debe ser capaz de interpretar el texto mediante una representación numérica de este. Una forma de hacerlo es utilizando un modelo llamado 'bag-of-words'. Este modelo simplemente cuenta la frecuencia de los tokens de palabras para cada correo electrónico y, de este modo, lo representa como un vector de estos conteos.

Función Assemble matrices

La función `assemble_bag()` ensambla un dataframe nuevo que contiene todas las palabras únicas encontradas en documentos de texo. Cuenta la frecuencia por palabra y luego regresa el dataframe nuevo.

In [51]:
```
def assemble_bag(data):
    used_tokens = []
    all_tokens = []

    for item in data:
        for token in item:
            if token in all_tokens:
                if token not in used_tokens:
                    used_tokens.append(token)
            else:
                all_tokens.append(token)

    df = pd.DataFrame(0, index = np.arange(len(data)), columns = used_tokens

    for i, item in enumerate(data):
```

```
        for token in item:
            if token in used_tokens:
                #####7) ESCRIBAN UNA LÍNEA DE CÓDIGO QUE INCREMENTE EN 1 EL
                df.loc[i, token] += 1
    return df
```

Poniendo todo junto para armar el Dataset

Se hacen todos los pasos de pre-procesamiento para armar nuestro dataset...

In [52]:
```
# Convertir todo a lower-case, truncar a maxtokens y truncar cada token a ma
EnronEmails = bodies_df.iloc[:,0].apply(tokenize)
EnronEmails = EnronEmails.apply(stop_word_removal)
EnronEmails = EnronEmails.apply(reg_expressions)
EnronEmails = EnronEmails.sample(Nsamp)

SpamEmails = fraud_bodies_df.iloc[:,0].apply(tokenize)
SpamEmails = SpamEmails.apply(stop_word_removal)
SpamEmails = SpamEmails.apply(reg_expressions)
SpamEmails = SpamEmails.sample(Nsamp)

raw_data = pd.concat([SpamEmails,EnronEmails], axis=0).values
```

In [53]:
```
print("El tamaño de los datos combinados es:")
print(raw_data.shape)
print("Los datos son:")
print(raw_data)

# crear las etiquetas correspondientes
Categories = ['spam','notspam']
header = ([1]*Nsamp)
header.extend(([0]*Nsamp))
```

```
El tamaño de los datos combinados es:
(2000,)
Los datos son:
[list(['saeed', 'ahmed', 'investmentesuite', 'c', 'city', 'tower', 'opposit
e', 'emirates', 'hotelsheikh', 'zayed', 'roadcdubaic', 'united', 'arab', 'em
irateseueaeeeapril', 'thcedear', 'friendc', '', 'as', 'read', 'thisc', 'i',
'dont', 'want', 'feel', 'sorry', 'mec', 'becausec', 'i', 'believe', 'everyon
e', 'die', 'somedaye', 'my', 'name', 'saeed', 'ahmed', 'merchant', 'in', 'du
baic', 'ueaeee', 'i', 'diagnosed', 'esophageal', 'cancere', ''])
 list(['dear', 'sir', '', 'it', 'heart', 'full', 'hope', 'i', 'write', 'see
k', 'help', 'respect', 'context', 'below', 'i', 'mrs', 'maryam', 'abacha',
'former', 'first', 'lady', 'former', 'military', 'head', 'state', 'nigeria',
'general', 'sani', 'abacha', 'whose', 'sudden', 'death', 'occurred', 'th',
'june', '', 'result', 'cardiac', 'arrest', 'heart', 'attack', 'seat', 'powe
r', 'i', 'doubt', 'capability', 'goodwill', 'assist', 'receiving', 'custod
y', 'for', 'safety', 'sum', 'us', 'million', 'willed', 'deposited', 'favou
r', 'late', 'husband', 'security', 'deposit', 'company', 'though', 'contac
t', 'assistance', 'anchored', 'personal', 'ecommendation', 'i', 'pray', 'und
erstanding', 'good', 'sincere', 'assistance', 'respond', 'message', 'hones
t', 'intensions', 'concern', 'this', 'money', 'currently', 'deposited', 'sec
urity', 'company', 'miscellaneous', 'awaiting', 'collection', 'according',
'agreement', 'entered', 'time', 'deposit', 'late', 'husband', 'security', 'c
ompany', 'time', 'deposit', 'collection', 'centre', 'ghana', 'as', 'legall
y', 'required', 'administration', 'late', 'husbands', 'properties'])
 list(['from', 'the', 'desk', 'of', 'ben', 'coroma', 'esqsenior', 'staff',
'in', 'filedepartmentafrican', 'development', 'bank', 'adbouagadougouburkina
faso', 'west', 'africacall', 'number', 'plane', 'crash', 'web', 'sitehttpnew
sbbccoukhiworldeuropestm', 'remittance', 'of', '', 'million', 'usa', 'dollar
s', 'confidential', 'is', 'thecasecompliment', 'of', 'the', 'seasonon', 'goo
d', 'day', 'i', 'ben', 'coroma', 'esqasenior', 'staff', 'filedepartment', 'a
frican', 'developent', 'bank', 'adbi', 'gotyour', 'contact', 'atourist', 'co
untry', 'country', 'transacted', 'abusiness', 'bank', 'when', 'searching',
'aforeign', 'partner', 'iassured', 'capability', 'reliability', 'champion',
'busineesopportunity', 'prayed', 'god', 'allah', 'youin', 'department', 'wed
iscovered', 'abandoned', 'sum', '', '', 'million', 'usa', 'dollars', 'fiftee
n', 'millionusa', 'dollars', '', 'in', 'account', 'belongs', 'one', 'ofour',
'foreign', 'customerwho', 'died', 'along', 'entire', 'family', 'monday', 's
t', 'july', '', 'planecrash', 'since', 'got', 'information', 'death', 'expec
tinghis', 'next', 'kin', 'come', 'claim', 'hismoney', 'releaseit', 'unless',
'somebody'])
 ...
 list(['todays', 'issuealert', 'sponsors', 'imageare', 'looking', 'invest',
'in', 'attract', 'investors', 'for', 'provide', 'services', 'understand', 'b
usiness', 'technology', 'dynamics', 'hottest', 'companies', 'emerging', 'ene
rgy', 'sector', 'attend', 'energy', 'venture', 'fair', 'june', '', '', '',
'', 'boston', 'ma', 'hear', 'ceos', '', 'hot', 'energy', 'companies', 'prese
nt', 'their', 'business', 'plans', 'complete', 'event', 'description', 'avai
lable', 'wwwenergyventurefaircom', 'call', 'nannette', 'mooney', '', '', 'ex
t', '', 'miss', 'last', 'week', 'catch', 'latest', 'deregulation', 'competit
ion', 'restructuring', 'developments', 'energy', 'industry', 'scientechs',
'issueswatch', 'correction', '', 'issuealert', 'the', 'us', 'senate', 'evenl
y', 'divided', 'between', 'democrats', 'republicans', 'since', 'november',
'', 'elections', 'seven', 'years', 'incorrectly', 'stated', 'column', 'we',
'apologize', 'misunderstanding', 'error', 'might', 'caused', 'imageimagema
y', '', 'edf', 'pursues', 'italys', 'montedison', 'lack', 'reciprocity', 'st
ill', 'factor', 'by', 'will', 'mcnamaradirector', 'electric', 'industry', 'a
```

```
nalysis', 'imagepower', 'giant', 'electricitde', 'france', 'edf', 'sent', 's
hock', 'waves', 'italian', 'financial', 'markets', 'government', 'ministrie
s', 'attempting', 'raise', 'its', 'stake', 'electric', 'firm', 'montedison',
'', 'percent', '', 'percent', 'however', 'romano', 'prodi', 'european', 'uni
on', 'commissioner', 'competition'])
 list(['kay', '', 'i', 'reviewed', 'captioned', 'my', 'comments', '', 'p',
'', '', 'the', 'date', 'a', 'july', '', '', '', 'also', 'p', 'i', 'sure',
'', 'damages', 'calculation', 'supposed', 'work', 'ld', 'calculation', 'sect
ion', '', 'interesting', 'reinstatement', 'provisions', '', 'id', 'love', 't
alk', 'sometime', 'find', 'why', 'i', 'assume', 'structure', 'require', 'hel
l', 'high', 'water', 'provisions', 'market', 'lds', 'i', 'saw', 'provision
s', 'delinking', 'provision', 'power', 'building', 'commissioning', 'projec
t', '', 'i', 'assume', 'herman', 'will', 'able', 'tell', 'delinking', 'lds',
'pass', 'muster', 'aa', 'purposes', 'marking', 'contract', 'getting', 'cas
h', 'flow', 'earnings', 'popthanksrose'])
 list(['did', 'chance', 'sign', 'justins', 'choice', 'indian', 'law', 'fir
m', 'forwarded', 'rob', 'wallsnaenron', '', '', 'am', 'wade', 'clineenron_de
velopment', '', 'am', '', 'to', 'sandeep', 'katwalaenron_developmentenron_de
velopment', 'cc', 'rob', 'wallsnaenronenron', 'subject', 're', 'clickpaperco
m', '', 'indian', 'legal', 'issuessandeep', 'find', 'firm', 'justin', 'discu
ssing', 'with', 'hopefully', 'good', 'firm', 'terms', 'quality', 'absence',
'conflicts', 'weve', 'done', 'in', 'india', 'date', 'let', 'knowi', 'hope',
'justin', 'contacted', 'someone', 'enron', 'legal', 'experience', 'india',
'before', 'hired', 'firm', 'hopefully', 'talked', 'first', 'common', 'courte
sy', 'maybe', 'talked', 'sarah', 'g', 'would', 'fine', 'also', 'she', 'gener
ally', 'aware', 'indian', 'firms', 'use', 'use', '', 'forwarded', 'wade', 'c
lineenron_development', '', '', 'pm', 'from', 'travis', 'mcculloughect', '',
'', 'pm', 'cstto', 'wade', 'clineenron_developmentenron_developmentcc', 'san
deep', 'katwalaenron_developmentenron_development', 'subject', 're', 'clickp
apercom', '', 'indian', 'legal', 'issues', 'wadethank', 'quick', 'responsea
s', 'i', 'suspected', 'i', 'slow', 'draw', 'justin', 'boyd', 'london', 'tea
m', 'already', 'directed', 'firm', 'india', 'start', 'looking', 'issuesill',
'forward', 'email', 'justin', 'well'])]
```

¡Estamos ahora listos para convertirlos en valores numéricos!!

Crear features y labels

In [54]:
```
# crear modelo bag-of-words
EnronSpamBag = assemble_bag(raw_data)
# la siguiente es la lista de palabras en nuestro modelo bag-of-words
predictors = [column for column in EnronSpamBag.columns]
EnronSpamBag # despliegue el modelo para el usuario
```

Out[54]:

	i	saeed	ahmed		former	abacha	heart	assistance	deposited	security
0	3	2	2	2	0	0	0	0	0	0
1	4	0	0	2	2	2	2	2	2	3
2	1	0	0	5	0	0	0	0	0	0
3	4	0	0	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
1995	0	0	0	0	0	0	0	0	0	0
1996	1	0	0	5	0	0	0	0	0	0
1997	0	0	0	13	0	0	0	0	0	0
1998	5	0	0	11	0	0	0	0	0	0
1999	2	0	0	11	0	0	0	0	0	0

2000 rows × 10793 columns

In [55]:

```
# hacer un shuffle inicial del raw data
def unison_shuffle_data(data, header):
    p = np.random.permutation(len(header))
    data = data[p,:]
    header = np.asarray(header)[p]
    return data, header
data, header = unison_shuffle_data(EnronSpamBag.values, header)

# dividir en conjuntos independientes
idx = int(0.7*data.shape[0])

# 70% de datos para training
train_x = data[:idx,:]
train_y = header[:idx]
# # restante 30% para testing
test_x = data[idx:,:]
test_y = header[idx:]

print("detalles de train_x/train_y, para asegurar que tengan la forma correc
print(len(train_x))
print(train_x)
print(train_y[:5])
print(len(train_y))
```

```
detalles de train_x/train_y, para asegurar que tengan la forma correcta:
1400
[[0 0 0 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 [2 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
[0 0 1 1 0]
1400
```

Como 70% de 2000 es 1400, !se ve bien! (para Nsamp=1000)

¡Sigamos!

Clasificador de Regresión Logística

In [56]:
```
from sklearn.linear_model import LogisticRegression

def fit(train_x,train_y):
    model = LogisticRegression()

    try:
        model.fit(train_x, train_y)
    except:
        pass
    return model

model = fit(train_x,train_y)
```

In [57]:
```
predicted_labels = model.predict(test_x)

# imprimir todos los labels para transparencia completa
print("DEBUG:: Los labels completos de regresión logística son:")
print(predicted_labels)
```

```
DEBUG:: Los labels completos de regresión logística son:
[1 1 0 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 1 0 1 1 0 0 1
 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0
 1 1 1 1 0 1 0 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 0 1 1 1 1 0 0 1 1 1 0 1 1 0 0
 0 0 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1
 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 0 0 1 1 1 0 0 1
 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 1
 0 0 1 1 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 0
 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0
 0 0 1 1 0 1 1 1 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 0 0 1 0 1 1 1
 1 0 0 1 1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0
 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 0 0 1 1 0 1 1 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0
 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 1
 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 0 0 1 0 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 0
 0 1 0 1 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1 0 0
 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 1
 1 1 0 1 0 1 1 0]
```

In [58]:
```
from sklearn.metrics import accuracy_score

acc_score = accuracy_score(test_y, predicted_labels)

print("DEBUG::El accuracy score de regresión logística es::")
print(acc_score)
```

```
DEBUG::El accuracy score de regresión logística es::
0.9933333333333333
```

Clasificador de Support Vector Machine

In [59]:
```
import time
from sklearn.svm import SVC # modelo de Support Vector Classification
```

In [60]:
```
# Crear un clasificador de soporte vectorial
clf = SVC(C=1, gamma="auto", probability=True)

# Ajustar el clasificador usando datos de entrenamiento
start_time = time.time()
clf.fit(train_x, train_y)
end_time = time.time()
print("Entrenar el Clasificador SVC tomó %3d segundos"%(end_time-start_time))

predicted_labels = clf.predict(test_x)
print("DEBUG::Las labels del Clasificador SVC son::")
print(predicted_labels)

acc_score = accuracy_score(test_y, predicted_labels)

print("DEBUG::El accuracy score del Clasificador SVC es::")
print(acc_score)
```

```
Entrenar el Clasificador SVC tomó  52 segundos
DEBUG::Las labels del Clasificador SVC son::
[1 1 0 1 0 1 0 0 1 1 1 1 0 0 1 1 0 0 1 0 0 1 0 1 1 1 1 0 1 0 1 0 1 1 0 0 1
 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0
 1 1 1 1 0 1 0 1 0 1 1 0 1 1 1 0 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0
 0 0 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 1 0 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1
 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 0 0 1 1 1 0 1 1
 1 0 0 0 0 1 0 1 1 0 1 1 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 1
 0 1 1 1 1 1 0 1 1 0 0 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 0
 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0
 0 0 1 1 0 1 0 1 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 0 0 1 0 1 1 1
 1 0 0 1 1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0
 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1
 1 0 1 1 1 1 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 0 0 1 0
 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 1 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 1 0 1 1
 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0 0 1 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 0
 0 1 1 1 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 1 1 0 1 1 1 0 0 1 1 1 0 0
 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 1 1 0 0 1
 1 1 0 1 0 1 1 0]
DEBUG::El accuracy score del Clasificador SVC es::
0.935
```

Random Forests

In [61]:
```
# Cargar el scikit's random forest classifier library
from sklearn.ensemble import RandomForestClassifier

# Create un Clasificador random forest. Por convención, clf significa 'Class
clf = RandomForestClassifier(n_jobs=1, random_state=0)

# Entrenar al clasificador para que tome las características de entrenamient
start_time = time.time()
clf.fit(train_x, train_y)
end_time = time.time()
print("Entrenar el Random Forest Classifier tomó %3d segundos"%(end_time-sta

predicted_labels = clf.predict(test_x)
print("DEBUG::Las etiquetas RF predecidas son::")
print(predicted_labels)

acc_score = accuracy_score(test_y, predicted_labels)

print("DEBUG::El RF testing accuracy score es::")
print(acc_score)
```

```
Entrenar el Random Forest Classifier tomó   2 segundos
DEBUG::Las etiquetas RF predecidas son::
[1 1 0 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 1 0 1 1 0 0 1
 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0
 1 1 1 1 0 1 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 1 1 1 0 0 1 1 1 0 1 1 0 0
 0 0 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1
 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 0 0 1 1 1 0 0 1
 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 1
 0 0 1 1 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 1 0 1 1 0 0 1 1 0 0 1 0
 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0
 0 0 1 1 0 1 1 1 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 0 0 1 0 1 1 1
 1 0 0 1 1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0
 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 0 0 1 1 0 1 1 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0
 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 1 0 0 1 1 0 1 1 0 0 1 1
 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 0
 0 1 0 1 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1 0 0
 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 1
 1 1 0 1 0 1 1 0]
DEBUG::El RF testing accuracy score es::
0.99
```

In [62]:
```
# Ahora, ajuste los parámetros sistemáticamente
from sklearn.model_selection import GridSearchCV

print("Available hyper-parameters for systematic tuning available with RF:")
print(clf.get_params())

## seleccione un subconjunto de parámetros a ajustar, y especifique grid par
param_grid = {
    'min_samples_leaf': [1, 2, 3],
    'min_samples_split': [2, 6, 10],
    'n_estimators': [10, 100, 1000]
}
grid_search = GridSearchCV(estimator = clf, param_grid = param_grid,
                           cv = 3, n_jobs = -1, verbose = 2)

# Ajuste la búsqueda de grid a los datos
grid_search.fit(train_x, train_y)

print("Los mejores parámetros encontrados:")
print(grid_search.best_params_)

print("La accuracy estimada es:")
acc_score = accuracy_score(test_y, grid_search.best_estimator_.predict(test_
print(acc_score)
```

```
Available hyper-parameters for systematic tuning available with RF:
{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gi
ni', 'max_depth': None, 'max_features': 'sqrt', 'max_leaf_nodes': None, 'max
_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_s
amples_split': 2, 'min_weight_fraction_leaf': 0.0, 'monotonic_cst': None, 'n
_estimators': 100, 'n_jobs': 1, 'oob_score': False, 'random_state': 0, 'verb
ose': 0, 'warm_start': False}
Fitting 3 folds for each of 27 candidates, totalling 81 fits
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=10; total tim
e=   0.6s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=10; total tim
e=   0.7s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=10; total tim
e=   0.7s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=10; total tim
e=   0.6s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=10; total tim
e=   0.6s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=10; total tim
e=   0.7s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=100; total ti
me=   3.5s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=100; total ti
me=   3.5s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=100; total ti
me=   3.6s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=100; total ti
me=   4.0s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=100; total ti
me=   4.3s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=10; total ti
me=   0.8s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=100; total ti
me=   4.2s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=10; total ti
me=   0.8s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=10; total ti
me=   0.8s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=100; total t
ime=   4.1s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=100; total t
ime=   4.2s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=100; total t
ime=   4.1s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total t
ime=  40.2s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total t
ime=  40.9s
[CV] END min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total t
ime=  41.5s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=10; total tim
e=   1.0s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=10; total tim
e=   0.9s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=10; total tim
e=   0.9s
```

```
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=1000; total t
ime=  40.4s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=1000; total t
ime=  41.4s
[CV] END min_samples_leaf=1, min_samples_split=6, n_estimators=1000; total t
ime=  41.5s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=100; total ti
me=   4.0s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=100; total ti
me=   4.0s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=100; total ti
me=   3.7s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=10; total tim
e=   0.6s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=10; total tim
e=   0.7s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=10; total tim
e=   0.9s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=1000; total
time=  39.8s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=100; total ti
me=   4.4s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=100; total ti
me=   4.2s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=100; total ti
me=   3.9s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=1000; total
time=  40.6s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=10; total ti
me=   1.2s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=10; total ti
me=   1.5s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=10; total ti
me=   1.4s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=100; total t
ime=   4.3s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=100; total t
ime=   4.4s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=100; total t
ime=   4.5s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=1000; total t
ime=  40.2s
[CV] END min_samples_leaf=1, min_samples_split=10, n_estimators=1000; total
time=  46.6s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=1000; total t
ime=  40.9s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=10; total tim
e=   0.8s
[CV] END min_samples_leaf=2, min_samples_split=2, n_estimators=1000; total t
ime=  41.7s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=10; total tim
e=   0.8s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=10; total tim
e=   0.9s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=100; total ti
me=   4.2s
```

```
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=100; total ti
me=   4.2s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=1000; total t
ime=  41.6s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=1000; total t
ime=  41.5s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=100; total ti
me=   3.1s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=10; total tim
e=   0.9s
[CV] END min_samples_leaf=2, min_samples_split=6, n_estimators=1000; total t
ime=  42.2s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=10; total tim
e=   0.8s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=10; total tim
e=   0.8s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=100; total ti
me=   4.0s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=100; total ti
me=   4.0s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=100; total ti
me=   3.8s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=1000; total
time=  38.7s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=1000; total
time=  36.4s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=10; total ti
me=   0.9s
[CV] END min_samples_leaf=2, min_samples_split=10, n_estimators=1000; total
time=  36.9s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=10; total ti
me=   0.9s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=10; total ti
me=   0.8s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=1000; total t
ime=  32.7s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=1000; total t
ime=  32.4s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=100; total t
ime=   3.6s
[CV] END min_samples_leaf=3, min_samples_split=2, n_estimators=1000; total t
ime=  33.0s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=100; total t
ime=   3.4s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=100; total t
ime=   2.7s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=1000; total t
ime=  30.6s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=1000; total t
ime=  29.3s
[CV] END min_samples_leaf=3, min_samples_split=6, n_estimators=1000; total t
ime=  26.0s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=1000; total
time=  18.2s
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=1000; total
time=  18.0s
```

```
[CV] END min_samples_leaf=3, min_samples_split=10, n_estimators=1000; total
time=  18.0s
Los mejores parámetros encontrados:
{'min_samples_leaf': 3, 'min_samples_split': 10, 'n_estimators': 100}
La accuracy estimada es:
0.9866666666666667
```

Máquinas Gradient Boosting

In [63]:
```
from sklearn.ensemble import GradientBoostingClassifier  # algoritmo GBM
from sklearn import metrics   #Funciones scklearn adicionales
from sklearn.model_selection import cross_val_score, GridSearchCV

def modelfit(alg, train_x, train_y, predictors, test_x, performCV=True, prir
    #Ajuste el algoritmo a los datos
    alg.fit(train_x, train_y)

    #Prediga conjunto de training:
    predictions = alg.predict(train_x)
    predprob = alg.predict_proba(train_x)[:,1]

    #Haga cross-validation:
    if performCV:
        cv_score = cross_val_score(alg, train_x, train_y, cv=cv_folds, scori

    #Imprima reporte de modelo:
    print("\nModel Report")
    print("Accuracy : %.4g" % metrics.accuracy_score(train_y,predictions))
    print("AUC Score (Train): %f" % metrics.roc_auc_score(train_y, predprob)

    if performCV:
        print("CV Score : Mean - %.7g | Std - %.7g | Min - %.7g | Max - %.7g

    #Imprimir Feature Importance:
    if printFeatureImportance:
        feat_imp = pd.Series(alg.feature_importances_, predictors).sort_valu
        feat_imp[:10].plot(kind='bar', title='Feature Importances')

    return alg.predict(test_x),alg.predict_proba(test_x)

gbm0 = GradientBoostingClassifier(random_state=10)

start_time = time.time()
test_predictions, test_probs = modelfit(gbm0, train_x, train_y, predictors,
end_time = time.time()

print("El entrenamiento del Gradient Boosting Classifier tomó %3d segundos"%

predicted_labels = test_predictions
print("DEBUG::Los labels predecidos de Gradient Boosting son::")
print(predicted_labels)

acc_score = accuracy_score(test_y, predicted_labels)
```

```
print("DEBUG::El testing accuracy score de Gradient Boosting es::")
print(acc_score)
```

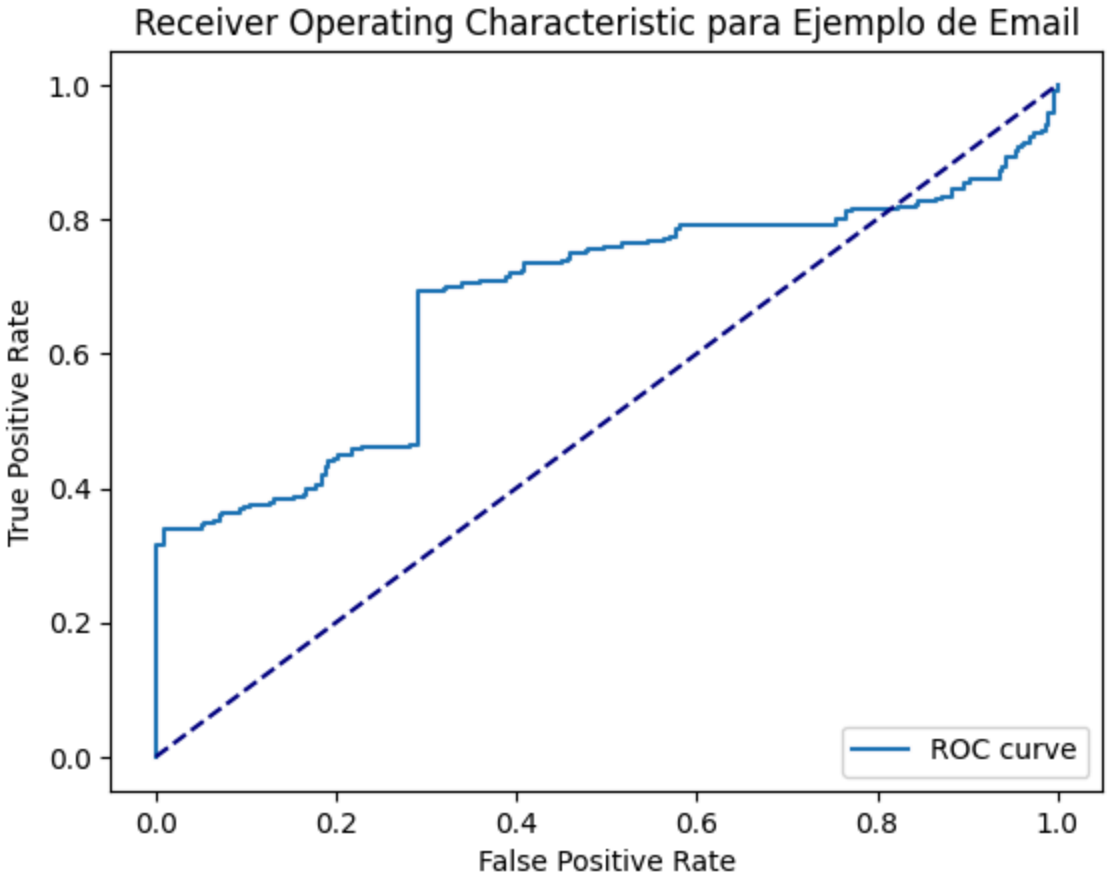
```
Model Report
Accuracy : 0.9971
AUC Score (Train): 0.998180
CV Score : Mean - 0.9942958 | Std - 0.002087056 | Min - 0.9902546 | Max - 0.
9959439
El entrenamiento del Gradient Boosting Classifier tomó 337 segundos
DEBUG::Los labels predecidos de Gradient Boosting son::
[1 1 0 1 0 0 0 0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 1 0 1 1 0 0 1
 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 0 1 0 0 1 1 0 0 0 0 1 0 1 1 0
 1 1 1 1 0 1 0 1 0 1 1 0 0 1 1 0 1 0 0 1 1 0 0 1 1 1 1 0 0 1 1 1 0 1 1 0 0
 0 0 1 1 1 0 0 0 0 1 1 0 1 1 1 0 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1
 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1 1 0 0 1 1 1 0 0 1
 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 1 0 1 0 1 0 1 1 0 1 0 1 0 1 1 0 0 0 0 0 0 1
 0 0 1 1 1 1 0 1 0 0 0 1 1 1 0 1 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 1 1 0 0 1 0
 0 0 0 0 0 1 0 1 0 0 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0 1 1 1 0 1 0 0 1 1 1 0 0
 0 0 1 1 0 1 0 1 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 1 1 1 1 0 0 0 1 0 1 1 1
 1 0 0 1 1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0
 0 1 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 1 0 0 1 0 0 1 1 0 1 1 1 1 1 0 1 0 0 0 1 1 1 0 0 1 0
 0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 0 1 0 0 1 0 0 1 1 0 0 1 1
 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 1 1 0 1 0 1 1 0 1 0 1 1 0 0 1 0 0 1 0 0 0
 0 1 0 1 1 0 0 1 0 0 1 1 1 0 1 0 1 1 0 1 1 1 1 0 1 1 0 1 0 1 0 0 1 1 1 0 0
 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 1
 0 1 0 1 0 1 1 0]
DEBUG::El testing accuracy score de Gradient Boosting es::
0.9833333333333333
```

In [64]:

```
# Haga una curva ROC
test_probs_max = [] # primero encontrar las probabilidades correspondientes
                    # a la clase más probable (máxima probabilidad)
for i in range(test_probs.shape[0]):
    test_probs_max.append(test_probs[i,test_y[i]])
len(test_probs_max)

# ahora, generar los datos de la curva
fpr, tpr, thresholds = metrics.roc_curve(test_y, np.array(test_probs_max))

# plot curve data
import matplotlib.pyplot as plt
fig,ax = plt.subplots()
plt.plot(fpr,tpr,label='ROC curve')
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic para Ejemplo de Email')
plt.legend(loc="lower right")
plt.show()
```



Now, save the image and make it downloadable, so you can use it in local documents

In [65]:
```
fig.savefig('ROC.eps', format='eps',bbox_inches='tight')
fig.savefig('ROC.pdf', format='pdf',bbox_inches='tight')
fig.savefig('ROC.png', format='png',bbox_inches='tight')
fig.savefig('ROC.svg', format='svg',bbox_inches='tight')
```

```
The PostScript backend does not support transparency; partially transparent
artists will be rendered opaque.
```

In [66]:
```
# Salvar .svg de ROC
from IPython.display import HTML
def create_download_link(title = "Download file", filename = "ROC.svg"):
    html = '<a href={filename}>{title}</a>'
    html = html.format(title=title,filename=filename)
    return HTML(html)

create_download_link(filename='ROC.svg')
```

Out[66]: Download file

Conclusion

Despues de probar diferentes modelos podemos ver el tiempo que tardaron en ser entrenados, algunos de estos modelos fueron rapidos mientras que otros fueron lentos, nuestro mejor accuracy fue de 0.9866 que fue el random forest, este modelo es bueno,

fue el que utilizamos en el reto pasado por los resultados que daba y por como se entrenaba, este modelo fue entranado rapido a comparacion de otros modelos que tambien consiguieron un accuracy score arriba de .9 pero la diferencia de tiempo en el entreno si fue significante.

In [ ]:
```
!jupyter nbconvert --to html 'baseline_clasificacion_spam_email.ipynb'
```