**PROGETTO CLASSIFICAZIONE DEL TESTO:** (Il progetto è diviso in 3 classificatori)

1. Classificatore del "sentiment"
2. Classificatore di email spam
3. Riassumere un testo

**Il progetto è ispirato a ciò che ho fatto durante il corso seguito sulla piattaforma Udemy: "Natural Language Processing", di ProfessionAI**

I modelli saranno addestrati in inglese per la scarsa quantità di dataset in italiano e librerie di Machine Learning che implementano la lingua italiana

1. **CLASSIFICAZIONE DEL "SENTIMENT"** (sentiment analysis), il modello che cercherà di capire se una frase è positiva o negativa

Inanzitutto bisogna importare il dataset sul quale deve essere addestrato il modello

```
!wget http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz
```

```
--2024-04-10 18:04:04--
http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz
Resolving ai.stanford.edu (ai.stanford.edu)... 171.64.68.10
Connecting to ai.stanford.edu (ai.stanford.edu)|171.64.68.10|:80...
connected.
HTTP request sent, awaiting response... 200 OK
Length: 84125825 (80M) [application/x-gzip]
Saving to: ‘aclImdb_v1.tar.gz.1’

aclImdb_v1.tar.gz.1 100%[===================>]  80.23M  12.7MB/s    in
13s

2024-04-10 18:04:18 (6.06 MB/s) - ‘aclImdb_v1.tar.gz.1’ saved
[84125825/84125825]
```

```
# estrazione del file compresso
!tar -xzf aclImdb_v1.tar.gz
```

Creazione di una funzione per leggere tutte le recensioni da tutti i files per poi ritornarle insieme al target corrispondente

```
from os import listdir
from sklearn.utils import shuffle


def get_xy(files_path, labels=["pos","neg"]):


    label_map = {labels[0]:1, labels[1]:0}

    reviews = []
```

```
    y = []

    for label in labels:
      path = files_path+label
      for file in listdir(path):
        review_file = open(path+"/"+file)
        review = review_file.read()

        reviews.append(review)
        y.append(label_map[label])

    # la funzione shuffle di sklearn ci permette di
    # mescolare più array allo stesso modo

    reviews, y = shuffle(reviews,y)

    return(reviews,y)
```

Utilizzo della funzione per ottenere le recensioni e il target in due liste

```
reviews_train, y_train = get_xy("aclImdb/train/")
reviews_test, y_test = get_xy("aclImdb/test/")

print("Prima recensione del set di test")
print(reviews_test[0])
print("Sentimeny: %d" % y_test[0])
```

```
Prima recensione del set di test
The action was episodic and there was no narrative thread to tie the
episodes together and move the story forward. The plot plods along.
With few exceptions (e.g., Graham Greene) the acting was uninspired,
and pedestrian at best. The actors seemed to have something on their
minds, other than the scene they were in. It is boring to observe a
man driving a car through the semi- desert country of this movie's
setting, whether he drives poorly or well. Such scenes are typical of
the level of tension in the video. So there was nothing about this
video to engage or draw the observer in, to make him or her care about
the characters and the out comes. I am doubly disappointed because I
rented this movie based on the reputations of the executive producer
(Redford) and the writer of the novel on which it was based
(Hillerman). I note that the jewel box reports that funding is
provided by PBS and the Corporation for Public Broadcasting, as well
as Carlton International. I would hope that this video was as
disappointing to them as it was to me and my wife, to the point that
they will not fund any more disasters coming from the same source.
Sentimeny: 0
```

Codifica delle recensioni ("bag of words")

```
from sklearn.feature_extraction.text import CountVectorizer

bow = CountVectorizer(max_features=5000)

bow_train = bow.fit_transform(reviews_train)
bow_test = bow.transform(reviews_test)

X_train = bow_train.toarray()
X_test = bow_test.toarray()

X_train.shape
```

```
(25000, 5000)
```

Standardizzazione degli array creati

```
from sklearn.preprocessing import StandardScaler

ss = StandardScaler()

X_train = ss.fit_transform(X_train)
X_test = ss.transform(X_test)
```

Creazione del modello e addestramento tramite regressione logistica

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(C=0.001)
lr.fit(X_train, y_train)
```

```
LogisticRegression(C=0.001)
```

Predizione e verifica del risultato tramite 2 parametri che misurano l'accuratezza del modello:

log loss e accuracy

```
from sklearn.metrics import accuracy_score, log_loss

train_pred = lr.predict(X_train)
train_pred_proba = lr.predict_proba(X_train)

train_accuracy = accuracy_score(y_train, train_pred)
train_loss = log_loss(y_train, train_pred_proba)

test_pred = lr.predict(X_test)
test_pred_proba = lr.predict_proba(X_test)

test_accuracy = accuracy_score(y_test, test_pred)
test_loss = log_loss(y_test, test_pred_proba)
```

```
print("Train Accuracy %.4f - Train Loss %.4f" % (train_accuracy,
train_loss))
print("Test Accuracy %.4f - Test Loss %.4f" % (test_accuracy,
test_loss))
```

```
Train Accuracy 0.9437 - Train Loss 0.1974
Test Accuracy 0.8775 - Test Loss 0.3126
```

il modello è piuttosto accurato (94% sui dati di addestramento e 87% sui dati di test)

**-> PROVA DEL MODELLO** (Analisi del sentiment [positivo o negativo])

```
# PRIMA RECENSIONE DA CLASSIFICARE (POSITIVA)
review = "This is the best movie I've ever seen"
prediction = lr.predict(bow.transform([review]))
if prediction[0] == 0: # se negativa modello riporta 0
  print("La recensione è negativa")
else: # altrimenti riporta 1
  print("La recensione è positiva")
```

```
La recensione è positiva
```

```
# SECONDA RECENSIONE DA CLASSIFICARE (NEGATIVA)
review = "This is the worst movie I've ever seen"
prediction = lr.predict(bow.transform([review]))
if prediction[0] == 0:
  print("La recensione è negativa")
else:
  print("La recensione è positiva")
```

```
La recensione è negativa
```

-> il modello ha riconosciuto con successo quale recensione era positiva e quale era negativa

1. **CLASSIFICAZIONE DI EMAIL SPAM** (spam classification)

Importazione librerie

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm
```

Lettura del dataset di addestramento

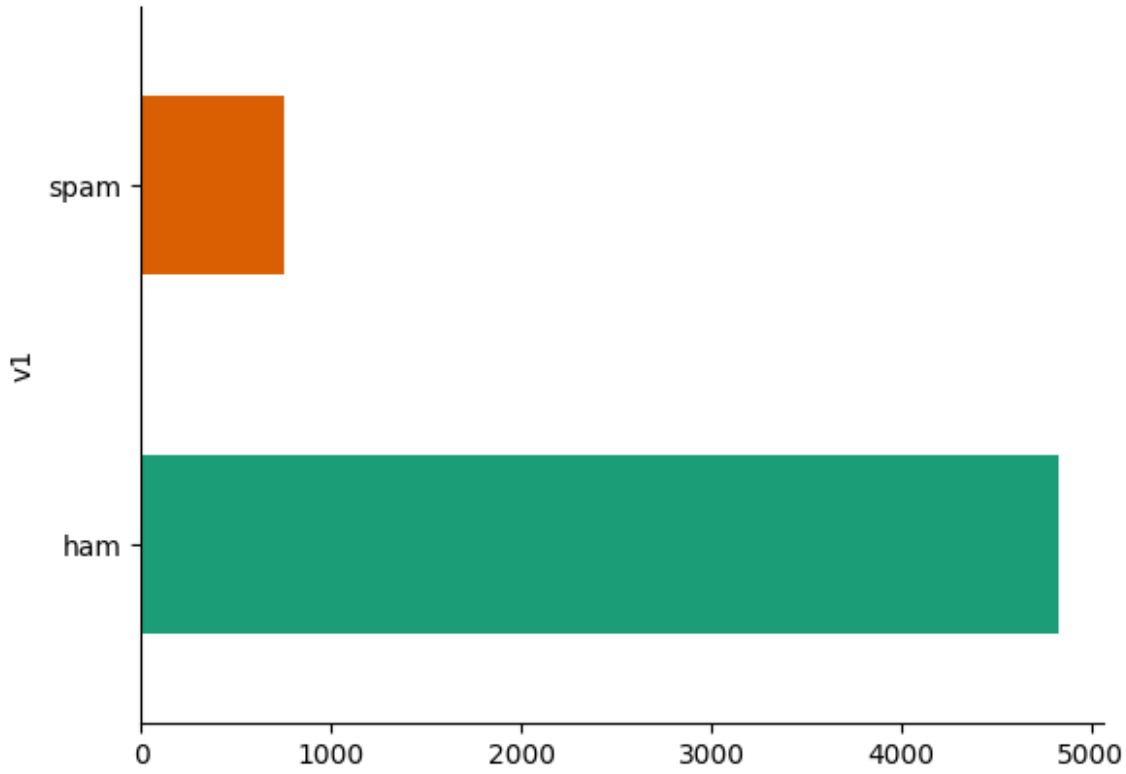
```
spam = pd.read_csv('/content/spam.csv')

spam.head()
```

{"summary":"{\n  \"name\": \"spam\",\n  \"rows\": 5572,\n  \"fields\":
[\n    {\n      \"column\": \"v1\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"spam\",\n          \"ham\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"v2\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 5169,\n
\"samples\": [\n          \"Did u download the fring app?\",\n
\"Pass dis to all ur contacts n see wat u get! Red;i'm in luv wid u.
Blue;u put a smile on my face. Purple;u r realy hot. Pink;u r so swt.
Orange;i thnk i lyk u. Green;i realy wana go out wid u. Yelow;i wnt u
bck. Black;i'm jealous of u. Brown;i miss you Nw plz giv me one
color\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Unnamed: 2\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 43,\n
\"samples\": [\n          \" GOD said\",\n          \" SHE SHUDVETOLD
U. DID URGRAN KNOW?NEWAY\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Unnamed: 3\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 10,\n
\"samples\": [\n          \" \\\\\\\"OH No! COMPETITION\\\\\\\". Who
knew\",\n          \" why to miss them\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Unnamed: 4\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"GNT:-)\\\"\",\n          \" one day these two will become FREINDS
FOREVER!\\\"\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"spam"}

```
# @title v1

from matplotlib import pyplot as plt
import seaborn as sns
spam.groupby('v1').size().plot(kind='barh',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```




Preparazione del dataset di addestramento tramite la creazione di label (etichette) e array

```
z = spam['v2'] # v2 = testo della mail
y = spam["v1"] # v1 = label (etichetta): spam o non spam (ham)
z_train, z_test,y_train, y_test = train_test_split(z,y,test_size =
0.2)
```

"Tokenizzazione": consiste nel dividere un testo in entità più piccole, chiamate token

```
cv = CountVectorizer()
features = cv.fit_transform(z_train)
```

Addestramento del modello

```
model = svm.SVC()
model.fit(features,y_train)

SVC()
```

Verifica sui dataset di test

```
features_test = cv.transform(z_test)
print(model.score(features_test,y_test))

0.9847533632286996
```

il modello è molto accurato (98%)

**-> PROVA DEL MODELLO** (Classificazione di una mail [spam o no])

```
# PROVA SU UNA MAIL SPAM
email = ["URGENT! You have won a 1 week FREE membership in our
å£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C
www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18"]
feature_test = cv.transform(email)
result = model.predict(feature_test)
print(f"La mail è {result[0]}")
```

```
La mail è spam
```

```
# PROVA SU UNA MAIL NON SPAM (HAM)
email = ["Please don't text me anymore. I have nothing else to say."]
feature_test = cv.transform(email)
result = model.predict(feature_test)
print(f"La mail è {result[0]}")
```

```
La mail è ham
```

-> il modello ha riconosciuto con successo la mail spam e la mail non spam (ham)

1. **RIASSUMERE UN TESTO**

Importazione dell'articolo da riassumere

```
!pip install newspaper3k
```

```
Requirement already satisfied: newspaper3k in
/usr/local/lib/python3.10/dist-packages (0.2.8)
Requirement already satisfied: beautifulsoup4>=4.4.1 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (4.12.3)
Requirement already satisfied: Pillow>=3.3.0 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (9.4.0)
Requirement already satisfied: PyYAML>=3.11 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (6.0.1)
Requirement already satisfied: cssselect>=0.9.2 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (1.2.0)
Requirement already satisfied: lxml>=3.6.0 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (4.9.4)
Requirement already satisfied: nltk>=3.2.1 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (3.8.1)
Requirement already satisfied: requests>=2.10.0 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (2.31.0)
Requirement already satisfied: feedparser>=5.2.1 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (6.0.11)
Requirement already satisfied: tldextract>=2.0.1 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (5.1.2)
Requirement already satisfied: feedfinder2>=0.0.4 in
```

```
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (0.0.4)
Requirement already satisfied: jieba3k>=0.35.1 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (0.35.1)
Requirement already satisfied: python-dateutil>=2.5.3 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (2.8.2)
Requirement already satisfied: tinysegmenter==0.3 in
/usr/local/lib/python3.10/dist-packages (from newspaper3k) (0.3)
Requirement already satisfied: soupsieve>1.2 in
/usr/local/lib/python3.10/dist-packages (from beautifulsoup4>=4.4.1-
>newspaper3k) (2.5)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-
packages (from feedfinder2>=0.0.4->newspaper3k) (1.16.0)
Requirement already satisfied: sgmllib3k in
/usr/local/lib/python3.10/dist-packages (from feedparser>=5.2.1-
>newspaper3k) (1.0.0)
Requirement already satisfied: click in
/usr/local/lib/python3.10/dist-packages (from nltk>=3.2.1-
>newspaper3k) (8.1.7)
Requirement already satisfied: joblib in
/usr/local/lib/python3.10/dist-packages (from nltk>=3.2.1-
>newspaper3k) (1.3.2)
Requirement already satisfied: regex>=2021.8.3 in
/usr/local/lib/python3.10/dist-packages (from nltk>=3.2.1-
>newspaper3k) (2023.12.25)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-
packages (from nltk>=3.2.1->newspaper3k) (4.66.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.10.0-
>newspaper3k) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.10.0-
>newspaper3k) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.10.0-
>newspaper3k) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.10/dist-packages (from requests>=2.10.0-
>newspaper3k) (2024.2.2)
Requirement already satisfied: requests-file>=1.4 in
/usr/local/lib/python3.10/dist-packages (from tldextract>=2.0.1-
>newspaper3k) (2.0.0)
Requirement already satisfied: filelock>=3.0.8 in
/usr/local/lib/python3.10/dist-packages (from tldextract>=2.0.1-
>newspaper3k) (3.13.3)

from newspaper import Article
from newspaper import Config

user_agent = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_5)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102
```

```
Safari/537.36'
config = Config()
config.browser_user_agent = user_agent
page =
Article("https://www.sciencedaily.com/releases/2021/08/210811162816.htm", config=config)
page.download()
page.parse()
print(page.text)
```

```
It is increasingly clear that the prolonged drought conditions, record-breaking heat, sustained wildfires, and frequent, more extreme storms experienced in recent years are a direct result of rising global temperatures brought on by humans' addition of carbon dioxide to the atmosphere. And a new MIT study on extreme climate events in Earth's ancient history suggests that today's planet may become more volatile as it continues to warm.

The study, appearing today in Science Advances, examines the paleoclimate record of the last 66 million years, during the Cenozoic era, which began shortly after the extinction of the dinosaurs. The scientists found that during this period, fluctuations in the Earth's climate experienced a surprising "warming bias." In other words, there were far more warming events -- periods of prolonged global warming, lasting thousands to tens of thousands of years -- than cooling events. What's more, warming events tended to be more extreme, with greater shifts in temperature, than cooling events.

The researchers say a possible explanation for this warming bias may lie in a "multiplier effect," whereby a modest degree of warming -- for instance from volcanoes releasing carbon dioxide into the atmosphere -- naturally speeds up certain biological and chemical processes that enhance these fluctuations, leading, on average, to still more warming.

Interestingly, the team observed that this warming bias disappeared about 5 million years ago, around the time when ice sheets started forming in the Northern Hemisphere. It's unclear what effect the ice has had on the Earth's response to climate shifts. But as today's Arctic ice recedes, the new study suggests that a multiplier effect may kick back in, and the result may be a further amplification of human-induced global warming.

"The Northern Hemisphere's ice sheets are shrinking, and could potentially disappear as a long-term consequence of human actions" says the study's lead author Constantin Arnscheidt, a graduate student in MIT's Department of Earth, Atmospheric and Planetary Sciences. "Our research suggests that this may make the Earth's climate fundamentally more susceptible to extreme, long-term global warming events such as those seen in the geologic past."
```

Arnscheidt's study co-author is Daniel Rothman, professor of geophysics at MIT, and co-founder and co-director of MIT's Lorenz Center.

## A volatile push

For their analysis, the team consulted large databases of sediments containing deep-sea benthic foraminifera -- single-celled organisms that have been around for hundreds of millions of years and whose hard shells are preserved in sediments. The composition of these shells is affected by the ocean temperatures as organisms are growing; the shells are therefore considered a reliable proxy for the Earth's ancient temperatures.

For decades, scientists have analyzed the composition of these shells, collected from all over the world and dated to various time periods, to track how the Earth's temperature has fluctuated over millions of years.

"When using these data to study extreme climate events, most studies have focused on individual large spikes in temperature, typically of a few degrees Celsius warming," Arnscheidt says. "Instead, we tried to look at the overall statistics and consider all the fluctuations involved, rather than picking out the big ones."

The team first carried out a statistical analysis of the data and observed that, over the last 66 million years, the distribution of global temperature fluctuations didn't resemble a standard bell curve, with symmetric tails representing an equal probability of extreme warm and extreme cool fluctuations. Instead, the curve was noticeably lopsided, skewed toward more warm than cool events. The curve also exhibited a noticeably longer tail, representing warm events that were more extreme, or of higher temperature, than the most extreme cold events.

"This indicates there's some sort of amplification relative to what you would otherwise have expected," Arnscheidt says. "Everything's pointing to something fundamental that's causing this push, or bias toward warming events."

"It's fair to say that the Earth system becomes more volatile, in a warming sense," Rothman adds.

## A warming multiplier

The team wondered whether this warming bias might have been a result of "multiplicative noise" in the climate-carbon cycle. Scientists have long understood that higher temperatures, up to a point, tend to speed up biological and chemical processes. Because the carbon cycle, which

```
is a key driver of long-term climate fluctuations, is itself composed of such processes, increases in temperature may lead to larger fluctuations, biasing the system towards extreme warming events.

In mathematics, there exists a set of equations that describes such general amplifying, or multiplicative effects. The researchers applied this multiplicative theory to their analysis to see whether the equations could predict the asymmetrical distribution, including the degree of its skew and the length of its tails.

In the end, they found that the data, and the observed bias toward warming, could be explained by the multiplicative theory. In other words, it's very likely that, over the last 66 million years, periods of modest warming were on average further enhanced by multiplier effects, such as the response of biological and chemical processes that further warmed the planet.

As part of the study, the researchers also looked at the correlation between past warming events and changes in Earth's orbit. Over hundreds of thousands of years, Earth's orbit around the sun regularly becomes more or less elliptical. But scientists have wondered why many past warming events appeared to coincide with these changes, and why these events feature outsized warming compared with what the change in Earth's orbit could have wrought on its own.

So, Arnscheidt and Rothman incorporated the Earth's orbital changes into the multiplicative model and their analysis of Earth's temperature changes, and found that multiplier effects could predictably amplify, on average, the modest temperature rises due to changes in Earth's orbit.

"Climate warms and cools in synchrony with orbital changes, but the orbital cycles themselves would predict only modest changes in climate," Rothman says. "But if we consider a multiplicative model, then modest warming, paired with this multiplier effect, can result in extreme events that tend to occur at the same time as these orbital changes."

"Humans are forcing the system in a new way," Arnscheidt adds. "And this study is showing that, when we increase temperature, we're likely going to interact with these natural, amplifying effects."

This research was supported, in part, by MIT's School of Science.
```

Importazione delle librerie

```
import spacy
from spacy.lang.en.stop_words import STOP_WORDS
from string import punctuation
from heapq import nlargest
```

Caricamento del modello dalla libreria spaCy

```
nlp = spacy.load('en_core_web_sm')
```

Codifica del testo (la libreria spaCy svolge la maggior parte delle operazioni in automatico)

```
doc= nlp(page.text)
```

Tokenizzazione

```
tokens=[token.text for token in doc]
```

Codifica delle parole del corpus di testo

```
word_frequencies={}
for word in doc:
  if word.text.lower() not in list(STOP_WORDS):
    if word.text.lower() not in punctuation:
      if word.text not in word_frequencies.keys():
        word_frequencies[word.text] = 1
      else:
        word_frequencies[word.text] += 1
max_frequency=max(word_frequencies.values())
for word in word_frequencies.keys():
  word_frequencies[word]=word_frequencies[word]/max_frequency
sentence_tokens= [sent for sent in doc.sents]
sentence_scores = {}
for sent in sentence_tokens:
  for word in sent:
    if word.text.lower() in word_frequencies.keys():
      if sent not in sentence_scores.keys():
        sentence_scores[sent]=word_frequencies[word.text.lower()]
      else:
        sentence_scores[sent]+=word_frequencies[word.text.lower()]
```

per = percentuale delle frasi dell'articolo che si desidera estrarre

```
per = 0.05

select_length=int(len(sentence_tokens)*per)
summary=nlargest(select_length,
sentence_scores,key=sentence_scores.get)
final_summary=[word.text for word in summary]
summary=''.join(final_summary)
```

**-> Prova del modello** (Riassumere un testo)

```
print(summary.replace(",", "\n"))
```

```
The researchers say a possible explanation for this warming bias may
lie in a "multiplier effect
" whereby a modest degree of warming -- for instance from volcanoes
releasing carbon dioxide into the atmosphere -- naturally speeds up
certain biological and chemical processes that enhance these
fluctuations
 leading
 on average
 to still more warming.

Because the carbon cycle
 which is a key driver of long-term climate fluctuations
 is itself composed of such processes
 increases in temperature may lead to larger fluctuations
 biasing the system towards extreme warming events.
```

-> l'articolo è stato riassunto con successo