```
#Install Packages
!pip install faiss-cpu
!pip install sentence-transformers

# import necessary libraries
import pandas as pd
pd.set_option('display.max_colwidth', 100)

df = pd.read_csv("sample_text.csv")
df.shape

(8, 2)

df


text  \
0                                         Meditation and yoga can
improve mental health
1                     Fruits, whole grains and vegetables helps
control blood pressure
2                                     These are the latest fashion
trends for this week
3                                     Vibrant color jeans for male are
becoming a trend
4                                                    The concert starts
at 7 PM tonight
5                      Navaratri dandiya program at Expo center in
Mumbai this october
6                                     Exciting vacation destinations
for your next trip
7  Maldives and Srilanka are gaining popularity in terms of low budget
vacation places

  category
0   Health
1   Health
2  Fashion
3  Fashion
4    Event
5    Event
6   Travel
7   Travel
```

Step 1 : Create source embeddings for the text column

```
from sentence_transformers import SentenceTransformer

C:\Users\dhava\AppData\Local\Programs\Python\Python310\lib\site-
packages\tqdm\auto.py:21: TqdmWarning: IProgress not found. Please
update jupyter and ipywidgets. See
```

```
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm

encoder = SentenceTransformer("all-mpnet-base-v2")
vectors = encoder.encode(df.text)

vectors.shape

(8, 768)

dim = vectors.shape[1]
dim

768
```

Step 2 : Build a FAISS Index for vectors

```
import faiss

index = faiss.IndexFlatL2(dim)
```

Step 3 : Normalize the source vectors (as we are using L2 distance to measure similarity) and add to the index

```
index.add(vectors)

index

<faiss.swigfaiss.IndexFlatL2; proxy of <Swig Object of type
'faiss::IndexFlatL2 *' at 0x00000252CF6123D0> >
```

Step 4 : Encode search text using same encorder and normalize the output vector

```
search_query = "I want to buy a polo t-shirt"
# search_query = "looking for places to visit during the holidays"
# search_query = "An apple a day keeps the doctor away"
vec = encoder.encode(search_query)
vec.shape

(768,)

import numpy as np
svec = np.array(vec).reshape(1,-1)
svec.shape

(1, 768)

# faiss.normalize_L2(svec)
```

Step 5: Search for similar vector in the FAISS index created

```
distances, I = index.search(new_vec, k=2)
distances
```

```
array([[1.3844836, 1.4039096]], dtype=float32)
```

```
I
```

```
array([[3, 2]], dtype=int64)
```

```
I.tolist()
```

```
[[3, 2]]
```

```
row_indices = I.tolist()[0]
row_indices
```

```
[3, 2]
```

```
df.loc[row_indices]
```

```
                                                text category
3  Vibrant color jeans for male are becoming a trend  Fashion
2  These are the latest fashion trends for this week  Fashion
```

```
search_query
```

```
'I want to buy a polo t-shirt'
```

You can see that the two results from the dataframe are similar to a search_query

Document Loaders In LangChain

TextLoader

```
from langchain.document_loaders import TextLoader

loader = TextLoader("nvda_news_1.txt")
loader.load()
```

```
[Document(page_content="The stock of NVIDIA Corp (NASDAQ:NVDA)
experienced a daily loss of -3.56% and a 3-month gain of 32.35%. With
an Earnings Per Share (EPS) (EPS) of $1.92, the question arises: is
the stock significantly overvalued? This article aims to provide a
detailed valuation analysis of NVIDIA, offering insights into its
financial strength, profitability, growth, and more. We invite you to
delve into this comprehensive analysis.\n\nCompany Overview\nWarning!
GuruFocus has detected 10 Warning Signs with NVDA. Click here to check
it out.\n\nNVDA 30-Year Financial Data\n\nThe intrinsic value of NVDA\
n\n\nNVIDIA Corp (NASDAQ:NVDA) is a leading designer of discrete
graphics processing units that enhance the experience on computing
platforms. The firm's chips are widely used in various end markets,
including PC gaming and data centers. In recent years, NVIDIA has
broadened its focus from traditional PC graphics applications such as
gaming to more complex and favorable opportunities, including
artificial intelligence and autonomous driving, leveraging the high-
performance capabilities of its products.\n\nCurrently, NVIDIA's stock
price stands at $418.01, significantly higher than the GF Value of
$310.28, indicating the stock might be overvalued. With a market cap
of $1 trillion, the valuation seems steep. The following analysis aims
to delve deeper into the company's value.\n\nIs NVIDIA's Stock
Significantly Overvalued? A Comprehensive Valuation Analysis\nIs
NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation
Analysis\nUnderstanding the GF Value\nThe GF Value is a unique measure
of the intrinsic value of a stock, calculated based on historical
trading multiples, a GuruFocus adjustment factor, and future business
performance estimates. If the stock price is significantly above the
GF Value Line, it is overvalued, and its future return is likely to be
poor. Conversely, if it is significantly below the GF Value Line, its
future return will likely be higher.\n\nAccording to GuruFocus Value
calculation, NVIDIA (NASDAQ:NVDA) appears to be significantly
overvalued. The stock's current price of $418.01 per share and the
market cap of $1 trillion further strengthen this assumption.\n\nGiven
that NVIDIA is significantly overvalued, the long-term return of its
stock is likely to be much lower than its future business growth.\n\
nIs NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation
Analysis\nIs NVIDIA's Stock Significantly Overvalued? A Comprehensive
Valuation Analysis\nLink: These companies may deliver higher future
returns at reduced risk.\n\nFinancial Strength of NVIDIA\nExamining
the financial strength of a company is crucial before investing in its
stock. Companies with poor financial strength pose a higher risk of
permanent loss. NVIDIA's cash-to-debt ratio of 1.27 is worse than
58.04% of companies in the Semiconductors industry. However, NVIDIA's
overall financial strength is 8 out of 10, indicating a strong
financial position.\n\nIs NVIDIA's Stock Significantly Overvalued? A
Comprehensive Valuation Analysis\nIs NVIDIA's Stock Significantly
Overvalued? A Comprehensive Valuation Analysis\nProfitability and
Growth\nConsistent profitability over the long term reduces the risk
for investors. NVIDIA, with its profitability ranking of 10 out of 10,
has been profitable for the past 10 years. The company's operating
margin of 17.37% ranks better than 76.5% of companies in the
Semiconductors industry.\n\nHowever, growth is a crucial factor in a
company's valuation. NVIDIA's growth ranks worse than 52.99% of
companies in the Semiconductors industry, with its 3-year average
revenue growth rate better than 87.88% of companies in the industry.\
n\nROIC vs WACC\nComparing a company's return on invested capital
(ROIC) to its weighted average cost of capital (WACC) is an effective
way to evaluate its profitability. Over the past 12 months, NVIDIA's
ROIC was 20.32 while its WACC was 16.74, suggesting that the company
is creating value for its shareholders.\n\nIs NVIDIA's Stock
Significantly Overvalued? A Comprehensive Valuation Analysis\nIs
NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation
Analysis\nConclusion\nIn conclusion, NVIDIA (NASDAQ:NVDA) appears to
be significantly overvalued. Despite its strong financial condition
and profitability, its growth ranks lower than 52.99% of companies in
the Semiconductors industry. To learn more about NVIDIA stock, you can
check out its 30-Year Financials here.\n\nTo find out the high quality
companies that may deliver above-average returns, please check out
GuruFocus High Quality Low Capex Screener.\n\nThis article first
appeared on GuruFocus.", metadata={'source': 'nvda_news_1.txt'})]
```

```python
type(loader)
```

```
langchain.document_loaders.text.TextLoader
```

```python
loader.file_path
```

```
'nvda_news_1.txt'
```

CSVLoader

```python
from langchain.document_loaders.csv_loader import CSVLoader

loader = CSVLoader(file_path="movies.csv")
data = loader.load()
data
```

```
[Document(page_content='movie_id: 101\ntitle: K.G.F: Chapter 2\
nindustry: Bollywood\nrelease_year: 2022\nimdb_rating: 8.4\nstudio:
Hombale Films\nlanguage_id: 3\nbudget: 1\nrevenue: 12.5\nunit:
Billions\ncurrency: INR', metadata={'source': 'movies.csv', 'row':
```

```
0}),
 Document(page_content='movie_id: 102\ntitle: Doctor Strange in the
Multiverse of Madness\nindustry: Hollywood\nrelease_year: 2022\
nimdb_rating: 7\nstudio: Marvel Studios\nlanguage_id: 5\nbudget: 200\
nrevenue: 954.8\nunit: Millions\ncurrency: USD', metadata={'source':
'movies.csv', 'row': 1}),
 Document(page_content='movie_id: 103\ntitle: Thor: The Dark World\
nindustry: Hollywood\nrelease_year: 2013\nimdb_rating: 6.8\nstudio:
Marvel Studios\nlanguage_id: 5\nbudget: 165\nrevenue: 644.8\nunit:
Millions\ncurrency: USD', metadata={'source': 'movies.csv', 'row':
2}),
 Document(page_content='movie_id: 104\ntitle: Thor: Ragnarok\
nindustry: Hollywood\nrelease_year: 2017\nimdb_rating: 7.9\nstudio:
Marvel Studios\nlanguage_id: 5\nbudget: 180\nrevenue: 854\nunit:
Millions\ncurrency: USD', metadata={'source': 'movies.csv', 'row':
3}),
 Document(page_content='movie_id: 105\ntitle: Thor: Love and Thunder\
nindustry: Hollywood\nrelease_year: 2022\nimdb_rating: 6.8\nstudio:
Marvel Studios\nlanguage_id: 5\nbudget: 250\nrevenue: 670\nunit:
Millions\ncurrency: USD', metadata={'source': 'movies.csv', 'row':
4}),
 Document(page_content='movie_id: 106\ntitle: Sholay\nindustry:
Bollywood\nrelease_year: 1975\nimdb_rating: 8.1\nstudio: United
Producers\nlanguage_id: 1\nbudget: Not Available\nrevenue: Not
Available\nunit: Not Available\ncurrency: Not Available',
metadata={'source': 'movies.csv', 'row': 5}),
 Document(page_content='movie_id: 107\ntitle: Dilwale Dulhania Le
Jayenge\nindustry: Bollywood\nrelease_year: 1995\nimdb_rating: 8\
nstudio: Yash Raj Films\nlanguage_id: 1\nbudget: 400\nrevenue: 2000\
nunit: Millions\ncurrency: INR', metadata={'source': 'movies.csv',
'row': 6}),
 Document(page_content='movie_id: 108\ntitle: 3 Idiots\nindustry:
Bollywood\nrelease_year: 2009\nimdb_rating: 8.4\nstudio: Vinod Chopra
Films\nlanguage_id: 1\nbudget: 550\nrevenue: 4000\nunit: Millions\
ncurrency: INR', metadata={'source': 'movies.csv', 'row': 7}),
 Document(page_content='movie_id: 109\ntitle: Kabhi Khushi Kabhie
Gham\nindustry: Bollywood\nrelease_year: 2001\nimdb_rating: 7.4\
nstudio: Dharma Productions\nlanguage_id: 1\nbudget: 390\nrevenue:
1360\nunit: Millions\ncurrency: INR', metadata={'source':
'movies.csv', 'row': 8})]
```

```
data[0]
```

```
Document(page_content='movie_id: 101\ntitle: K.G.F: Chapter 2\
nindustry: Bollywood\nrelease_year: 2022\nimdb_rating: 8.4\nstudio:
Hombale Films\nlanguage_id: 3\nbudget: 1\nrevenue: 12.5\nunit:
Billions\ncurrency: INR', metadata={'source': 'movies.csv', 'row': 0})
```

```
loader = CSVLoader(file_path="movies.csv", source_column="title")
data = loader.load()
data
```

```
[Document(page_content='movie_id: 101\ntitle: K.G.F: Chapter 2\
nindustry: Bollywood\nrelease_year: 2022\nimdb_rating: 8.4\nstudio:
Hombale Films\nlanguage_id: 3\nbudget: 1\nrevenue: 12.5\nunit:
Billions\ncurrency: INR', metadata={'source': 'K.G.F: Chapter 2',
'row': 0}),
 Document(page_content='movie_id: 102\ntitle: Doctor Strange in the
Multiverse of Madness\nindustry: Hollywood\nrelease_year: 2022\
nimdb_rating: 7\nstudio: Marvel Studios\nlanguage_id: 5\nbudget: 200\
nrevenue: 954.8\nunit: Millions\ncurrency: USD', metadata={'source':
'Doctor Strange in the Multiverse of Madness', 'row': 1}),
 Document(page_content='movie_id: 103\ntitle: Thor: The Dark World\
nindustry: Hollywood\nrelease_year: 2013\nimdb_rating: 6.8\nstudio:
Marvel Studios\nlanguage_id: 5\nbudget: 165\nrevenue: 644.8\nunit:
Millions\ncurrency: USD', metadata={'source': 'Thor: The Dark World',
'row': 2}),
 Document(page_content='movie_id: 104\ntitle: Thor: Ragnarok\
nindustry: Hollywood\nrelease_year: 2017\nimdb_rating: 7.9\nstudio:
Marvel Studios\nlanguage_id: 5\nbudget: 180\nrevenue: 854\nunit:
Millions\ncurrency: USD', metadata={'source': 'Thor: Ragnarok', 'row':
3}),
 Document(page_content='movie_id: 105\ntitle: Thor: Love and Thunder\
nindustry: Hollywood\nrelease_year: 2022\nimdb_rating: 6.8\nstudio:
Marvel Studios\nlanguage_id: 5\nbudget: 250\nrevenue: 670\nunit:
Millions\ncurrency: USD', metadata={'source': 'Thor: Love and
Thunder', 'row': 4}),
 Document(page_content='movie_id: 106\ntitle: Sholay\nindustry:
Bollywood\nrelease_year: 1975\nimdb_rating: 8.1\nstudio: United
Producers\nlanguage_id: 1\nbudget: Not Available\nrevenue: Not
Available\nunit: Not Available\ncurrency: Not Available',
metadata={'source': 'Sholay', 'row': 5}),
 Document(page_content='movie_id: 107\ntitle: Dilwale Dulhania Le
Jayenge\nindustry: Bollywood\nrelease_year: 1995\nimdb_rating: 8\
nstudio: Yash Raj Films\nlanguage_id: 1\nbudget: 400\nrevenue: 2000\
nunit: Millions\ncurrency: INR', metadata={'source': 'Dilwale Dulhania
Le Jayenge', 'row': 6}),
 Document(page_content='movie_id: 108\ntitle: 3 Idiots\nindustry:
Bollywood\nrelease_year: 2009\nimdb_rating: 8.4\nstudio: Vinod Chopra
Films\nlanguage_id: 1\nbudget: 550\nrevenue: 4000\nunit: Millions\
ncurrency: INR', metadata={'source': '3 Idiots', 'row': 7}),
 Document(page_content='movie_id: 109\ntitle: Kabhi Khushi Kabhie
Gham\nindustry: Bollywood\nrelease_year: 2001\nimdb_rating: 7.4\
nstudio: Dharma Productions\nlanguage_id: 1\nbudget: 390\nrevenue:
1360\nunit: Millions\ncurrency: INR', metadata={'source': 'Kabhi
Khushi Kabhie Gham', 'row': 8})]
```

```
data[0].page_content
```

```
'movie_id: 101\ntitle: K.G.F: Chapter 2\nindustry: Bollywood\
nrelease_year: 2022\nimdb_rating: 8.4\nstudio: Hombale Films\
nlanguage_id: 3\nbudget: 1\nrevenue: 12.5\nunit: Billions\ncurrency:
INR'

data[0].metadata

{'source': 'K.G.F: Chapter 2', 'row': 0}
```

UnstructuredURLLoader

UnstructuredURLLoader of Langchain internally uses unstructured python library to load the content from url's

https://unstructured-io.github.io/unstructured/introduction.html

https://pypi.org/project/unstructured/#description

```
#installing necessary libraries, libmagic is used for file type
detection
!pip3 install unstructured libmagic python-magic python-magic-bin

Requirement already satisfied: unstructured in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (0.9.2)
Requirement already satisfied: libmagic in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (1.0)
Requirement already satisfied: python-magic in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (0.4.27)
Requirement already satisfied: python-magic-bin in c:\users\dhava\
appdata\local\programs\python\python310\lib\site-packages (0.4.14)
Requirement already satisfied: tabulate in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (from unstructured)
(0.9.0)
Requirement already satisfied: requests in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (from unstructured)
(2.31.0)
Requirement already satisfied: nltk in c:\users\dhava\appdata\local\
programs\python\python310\lib\site-packages (from unstructured)
(3.8.1)
Requirement already satisfied: chardet in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (from unstructured)
(5.2.0)
Requirement already satisfied: filetype in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (from unstructured)
(1.2.0)
Requirement already satisfied: lxml in c:\users\dhava\appdata\local\
programs\python\python310\lib\site-packages (from unstructured)
(4.9.3)
Requirement already satisfied: click in c:\users\dhava\appdata\local\
programs\python\python310\lib\site-packages (from nltk->unstructured)
```

```
(8.1.3)
Requirement already satisfied: tqdm in c:\users\dhava\appdata\local\
programs\python\python310\lib\site-packages (from nltk->unstructured)
(4.65.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\dhava\
appdata\local\programs\python\python310\lib\site-packages (from nltk-
>unstructured) (2023.8.8)
Requirement already satisfied: joblib in c:\users\dhava\appdata\local\
programs\python\python310\lib\site-packages (from nltk->unstructured)
(1.3.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\dhava\
appdata\local\programs\python\python310\lib\site-packages (from
requests->unstructured) (2.0.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dhava\
appdata\local\programs\python\python310\lib\site-packages (from
requests->unstructured) (2023.5.7)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (from requests-
>unstructured) (3.4)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\
dhava\appdata\local\programs\python\python310\lib\site-packages (from
requests->unstructured) (3.1.0)
Requirement already satisfied: colorama in c:\users\dhava\appdata\
local\programs\python\python310\lib\site-packages (from click->nltk-
>unstructured) (0.4.6)


[notice] A new release of pip is available: 23.0.1 -> 23.2.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```python
from langchain.document_loaders import UnstructuredURLLoader

loader = UnstructuredURLLoader(
    urls = [
        "https://www.moneycontrol.com/news/business/banks/hdfc-bank-
re-appoints-sanmoy-chakrabarti-as-chief-risk-officer-11259771.html",
        "https://www.moneycontrol.com/news/business/markets/market-
corrects-post-rbi-ups-inflation-forecast-icrr-bet-on-these-top-10-
rate-sensitive-stocks-ideas-11142611.html"
    ]
)

data = loader.load()
len(data)
```

```
2
```

```python
data[0].page_content[0:100]
```

```
'English\n\nHindi\n\nGujarati\n\nSpecials\n\nTrending Stocks\n\nIRFC\
xa0INE053F01010, IRFC, 543257\n\nTata Power\xa0INE'
```

```
data[0].metadata
```

```
{'source': 'https://www.moneycontrol.com/news/business/banks/hdfc-
bank-re-appoints-sanmoy-chakrabarti-as-chief-risk-officer-
11259771.html'}
```

Text Splitters

Why do we need text splitters in first place?

LLM's have token limits. Hence we need to split the text which can be large into small chunks so that each chunk size is under the token limit. There are various text splitter classes in langchain that allows us to do this.

```
# Taking some random text from wikipedia

text = """Interstellar is a 2014 epic science fiction film co-written,
directed, and produced by Christopher Nolan.
It stars Matthew McConaughey, Anne Hathaway, Jessica Chastain, Bill
Irwin, Ellen Burstyn, Matt Damon, and Michael Caine.
Set in a dystopian future where humanity is embroiled in a
catastrophic blight and famine, the film follows a group of astronauts
who travel through a wormhole near Saturn in search of a new home for
humankind.

Brothers Christopher and Jonathan Nolan wrote the screenplay, which
had its origins in a script Jonathan developed in 2007 and was
originally set to be directed by Steven Spielberg.
Kip Thorne, a Caltech theoretical physicist and 2017 Nobel laureate in
Physics,[4] was an executive producer, acted as a scientific
consultant, and wrote a tie-in book, The Science of Interstellar.
Cinematographer Hoyte van Hoytema shot it on 35 mm movie film in the
Panavision anamorphic format and IMAX 70 mm. Principal photography
began in late 2013 and took place in Alberta, Iceland, and Los
Angeles.
Interstellar uses extensive practical and miniature effects, and the
company Double Negative created additional digital effects.

Interstellar premiered in Los Angeles on October 26, 2014. In the
United States, it was first released on film stock, expanding to
venues using digital projectors. The film received generally positive
reviews from critics and grossed over $677 million worldwide ($715
million after subsequent re-releases), making it the tenth-highest-
grossing film of 2014.
It has been praised by astronomers for its scientific accuracy and
portrayal of theoretical astrophysics.[5][6][7] Interstellar was
nominated for five awards at the 87th Academy Awards, winning Best
Visual Effects, and received numerous other accolades."""
```

Manual approach of splitting the text into chunks

```python
# Say LLM token limit is 100, in that case we can do simple thing such as this

text[0:100]
```

```
'Interstellar is a 2014 epic science fiction film co-written, directed, and produced by Christopher N'
```

```python
# Well but we want complete words and want to do this for entire text, may be we can use Python's split funciton

words = text.split(" ")
len(words)
```

```
264
```

```python
chunks = []

s = ""
for word in words:
    s += word + " "
    if len(s)>200:
        chunks.append(s)
        s = ""

chunks.append(s)
```

```python
chunks[:2]
```

```
['Interstellar is a 2014 epic science fiction film co-written, directed, and produced by Christopher Nolan. \nIt stars Matthew McConaughey, Anne Hathaway, Jessica Chastain, Bill Irwin, Ellen Burstyn, Matt ',
 'Damon, and Michael Caine. \nSet in a dystopian future where humanity is embroiled in a catastrophic blight and famine, the film follows a group of astronauts who travel through a wormhole near Saturn in ']
```

Splitting data into chunks can be done in native python but it is a tidious process. Also if necessary, you may need to experiment with various delimiters in an iterative manner to ensure that each chunk does not exceed the token length limit of the respective LLM.

Langchain provides a better way through text splitter classes.

Using Text Splitter Classes from Langchain

CharacterTextSplitter

```python
from langchain.text_splitter import CharacterTextSplitter

splitter = CharacterTextSplitter(
    separator = "\n",
    chunk_size=200,
    chunk_overlap=0
)

chunks = splitter.split_text(text)
len(chunks)
```

```
Created a chunk of size 210, which is longer than the specified 200
Created a chunk of size 208, which is longer than the specified 200
Created a chunk of size 358, which is longer than the specified 200
```

```
9
```

```
for chunk in chunks:
    print(len(chunk))
```

```
105
120
210
181
197
207
128
357
253
```

As you can see, all though we gave 200 as a chunk size since the split was based on \n, it ended up creating chunks that are bigger than size 200.

Another class from Langchain can be used to recursively split the text based on a list of separators. This class is RecursiveTextSplitter. Let's see how it works

RecursiveTextSplitter

```
text
```

```
'Interstellar is a 2014 epic science fiction film co-written,
directed, and produced by Christopher Nolan. \nIt stars Matthew
McConaughey, Anne Hathaway, Jessica Chastain, Bill Irwin, Ellen
Burstyn, Matt Damon, and Michael Caine. \nSet in a dystopian future
where humanity is embroiled in a catastrophic blight and famine, the
film follows a group of astronauts who travel through a wormhole near
Saturn in search of a new home for humankind.\n\nBrothers Christopher
and Jonathan Nolan wrote the screenplay, which had its origins in a
script Jonathan developed in 2007 and was originally set to be
directed by Steven Spielberg. \nKip Thorne, a Caltech theoretical
physicist and 2017 Nobel laureate in Physics,[4] was an executive
producer, acted as a scientific consultant, and wrote a tie-in book,
The Science of Interstellar. \nCinematographer Hoyte van Hoytema shot
it on 35 mm movie film in the Panavision anamorphic format and IMAX 70
```

```
mm. Principal photography began in late 2013 and took place in
Alberta, Iceland, and Los Angeles. \nInterstellar uses extensive
practical and miniature effects, and the company Double Negative
created additional digital effects.\n\nInterstellar premiered in Los
Angeles on October 26, 2014. In the United States, it was first
released on film stock, expanding to venues using digital projectors.
The film received generally positive reviews from critics and grossed
over $677 million worldwide ($715 million after subsequent re-
releases), making it the tenth-highest-grossing film of 2014. \nIt has
been praised by astronomers for its scientific accuracy and portrayal
of theoretical astrophysics.[5][6][7] Interstellar was nominated for
five awards at the 87th Academy Awards, winning Best Visual Effects,
and received numerous other accolades.'
```

```
from langchain.text_splitter import RecursiveCharacterTextSplitter

r_splitter = RecursiveCharacterTextSplitter(
    separators = ["\n\n", "\n", " "],  # List of separators based on requirement (defaults to ["\n\n", "\n", " "])
    chunk_size = 200,  # size of each chunk created
    chunk_overlap  = 0,  # size of  overlap between chunks in order to maintain the context
    length_function = len  # Function to calculate size, currently we are using "len" which denotes length of string however you can pass any token counter)
)

chunks = r_splitter.split_text(text)

for chunk in chunks:
    print(len(chunk))
```

```
105
120
199
10
181
197
198
8
128
191
165
198
54
```

Let's understand how exactly it formed these chunks

```
first_split = text.split("\n\n")[0]
first_split
```

```
'Interstellar is a 2014 epic science fiction film co-written,
directed, and produced by Christopher Nolan. \nIt stars Matthew
McConaughey, Anne Hathaway, Jessica Chastain, Bill Irwin, Ellen
Burstyn, Matt Damon, and Michael Caine. \nSet in a dystopian future
where humanity is embroiled in a catastrophic blight and famine, the
film follows a group of astronauts who travel through a wormhole near
Saturn in search of a new home for humankind.'
```

```
len(first_split)
```

```
439
```

Recursive text splitter uses a list of separators, i.e. separators = ["\n\n", "\n", "."]

So now it will first split using \n\n and then if the resulting chunk size is greater than the chunk_size parameter which is 200 in our case, then it will use the next separator which is \n

```
second_split = first_split.split("\n")
second_split
```

```
['Interstellar is a 2014 epic science fiction film co-written,
directed, and produced by Christopher Nolan. ',
 'It stars Matthew McConaughey, Anne Hathaway, Jessica Chastain, Bill
Irwin, Ellen Burstyn, Matt Damon, and Michael Caine. ',
 'Set in a dystopian future where humanity is embroiled in a
catastrophic blight and famine, the film follows a group of astronauts
who travel through a wormhole near Saturn in search of a new home for
humankind.']
```

```
for split in second_split:
    print(len(split))
```

```
106
121
210
```

Third split exceeds chunk size 200. Now it will further try to split that using the third separator which is ' ' (space)

```
second_split[2]
```

```
'Set in a dystopian future where humanity is embroiled in a
catastrophic blight and famine, the film follows a group of astronauts
who travel through a wormhole near Saturn in search of a new home for
humankind.'
```

When you split this using space (i.e. second_split[2].split(" ")), it will separate out each word and then it will merge those chunks such that their size is close to 200

The stock of NVIDIA Corp (NASDAQ:NVDA) experienced a daily loss of -3.56% and a 3-month gain of 32.35%. With an Earnings Per Share (EPS) (EPS) of $1.92, the question arises: is the stock significantly overvalued? This article aims to provide a detailed valuation analysis of NVIDIA, offering insights into its financial strength, profitability, growth, and more. We invite you to delve into this comprehensive analysis.

Company Overview

Warning! GuruFocus has detected 10 Warning Signs with NVDA. Click here to check it out.

NVDA 30-Year Financial Data

The intrinsic value of NVDA

NVIDIA Corp (NASDAQ:NVDA) is a leading designer of discrete graphics processing units that enhance the experience on computing platforms. The firm's chips are widely used in various end markets, including PC gaming and data centers. In recent years, NVIDIA has broadened its focus from traditional PC graphics applications such as gaming to more complex and favorable opportunities, including artificial intelligence and autonomous driving, leveraging the high-performance capabilities of its products.

Currently, NVIDIA's stock price stands at $418.01, significantly higher than the GF Value of $310.28, indicating the stock might be overvalued. With a market cap of $1 trillion, the valuation seems steep. The following analysis aims to delve deeper into the company's value.

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Understanding the GF Value

The GF Value is a unique measure of the intrinsic value of a stock, calculated based on historical trading multiples, a GuruFocus adjustment factor, and future business performance estimates. If the stock price is significantly above the GF Value Line, it is overvalued, and its future return is likely to be poor. Conversely, if it is significantly below the GF Value Line, its future return will likely be higher.

According to GuruFocus Value calculation, NVIDIA (NASDAQ:NVDA) appears to be significantly overvalued. The stock's current price of $418.01 per share and the market cap of $1 trillion further strengthen this assumption.

Given that NVIDIA is significantly overvalued, the long-term return of its stock is likely to be much lower than its future business growth.

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Link: These companies may deliver higher future returns at reduced risk.

Financial Strength of NVIDIA

Examining the financial strength of a company is crucial before investing in its stock. Companies with poor financial strength pose a higher risk of permanent loss. NVIDIA's cash-to-debt ratio of 1.27 is worse than 58.04% of companies in the Semiconductors industry. However, NVIDIA's overall financial strength is 8 out of 10, indicating a strong financial position.

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Profitability and Growth

Consistent profitability over the long term reduces the risk for investors. NVIDIA, with its profitability ranking of 10 out of 10, has been profitable for the past 10 years. The company's operating margin of 17.37% ranks better than 76.5% of companies in the Semiconductors industry.

However, growth is a crucial factor in a company's valuation. NVIDIA's growth ranks worse than 52.99% of companies in the Semiconductors industry, with its 3-year average revenue growth rate better than 87.88% of companies in the industry.

ROIC vs WACC

Comparing a company's return on invested capital (ROIC) to its weighted average cost of capital (WACC) is an effective way to evaluate its profitability. Over the past 12 months, NVIDIA's ROIC was 20.32 while its WACC was 16.74, suggesting that the company is creating value for its shareholders.

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Is NVIDIA's Stock Significantly Overvalued? A Comprehensive Valuation Analysis

Conclusion

In conclusion, NVIDIA (NASDAQ:NVDA) appears to be significantly overvalued. Despite its strong financial condition and profitability, its growth ranks lower than 52.99% of companies in the Semiconductors industry. To learn more about NVIDIA stock, you can check out its 30-Year Financials here.

To find out the high quality companies that may deliver above-average returns, please check out GuruFocus High Quality Low Capex Screener.

This article first appeared on GuruFocus.