```
import os
os.environ["OPENAI_API_KEY"] = "YOUR-API-KEY"
```

```
!pip install langchain
!pip install openai
!pip install python-dotenv
```

```
Collecting langchain
  Downloading langchain-0.1.12-py3-none-any.whl (809 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 809.1/809.1 kB 5.1 MB/s eta 0:00:00
Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.28)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.9.3)
Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
Collecting dataclasses-json<0.7,>=0.5.7 (from langchain)
  Downloading dataclasses_json-0.6.4-py3-none-any.whl (28 kB)
Collecting jsonpatch<2.0,>=1.33 (from langchain)
  Downloading jsonpatch-1.33-py2.py3-none-any.whl (12 kB)
Collecting langchain-community<0.1,>=0.0.28 (from langchain)
  Downloading langchain_community-0.0.28-py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 31.6 MB/s eta 0:00:00
Collecting langchain-core<0.2.0,>=0.1.31 (from langchain)
  Downloading langchain_core-0.1.31-py3-none-any.whl (258 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 258.8/258.8 kB 19.9 MB/s eta 0:00:00
Collecting langchain-text-splitters<0.1,>=0.0.1 (from langchain)
  Downloading langchain_text_splitters-0.0.1-py3-none-any.whl (21 kB)
Collecting langsmith<0.2.0,>=0.1.17 (from langchain)
  Downloading langsmith-0.1.25-py3-none-any.whl (67 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.6/67.6 kB 6.0 MB/s eta 0:00:00
Requirement already satisfied: numpy<2,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (1.25.2)
Requirement already satisfied: pydantic<3,>=1 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.6.3)
Requirement already satisfied: requests<3,>=2 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.31.0)
Requirement already satisfied: tenacity<9.0.0,>=8.1.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (8.2.3)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (23.2
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp<4.0.0,>=3.8.3->langchain) (1.9
Collecting marshmallow<4.0.0,>=3.18.0 (from dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading marshmallow-3.21.1-py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.4/49.4 kB 5.7 MB/s eta 0:00:00
Collecting typing-inspect<1,>=0.4.0 (from dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading typing_inspect-0.9.0-py3-none-any.whl (8.8 kB)
Collecting jsonpointer>=1.9 (from jsonpatch<2.0,>=1.33->langchain)
  Downloading jsonpointer-2.4-py2.py3-none-any.whl (7.8 kB)
Requirement already satisfied: anyio<5,>=3 in /usr/local/lib/python3.10/dist-packages (from langchain-core<0.2.0,>=0.1.31->langchain)
Collecting packaging<24.0,>=23.2 (from langchain-core<0.2.0,>=0.1.31->langchain)
  Downloading packaging-23.2-py3-none-any.whl (53 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.0/53.0 kB 5.3 MB/s eta 0:00:00
Collecting orjson<4.0.0,>=3.9.14 (from langsmith<0.2.0,>=0.1.17->langchain)
  Downloading orjson-3.9.15-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (138 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 138.5/138.5 kB 5.1 MB/s eta 0:00:00
Requirement already satisfied: annotated-types>=0.4.0 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (0.6
Requirement already satisfied: pydantic-core==2.16.3 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (2.16
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages (from pydantic<3,>=1->langchain) (4
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (3
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<3,>=2->langchain) (2024.2
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.10/dist-packages (from SQLAlchemy<3,>=1.4->langchain) (3.0.3
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3->langchain-core<0.2.0,>=0.1.3
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from anyio<5,>=3->langchain-core<0.2.0,>=0.1
Collecting mypy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-json<0.7,>=0.5.7->langchain)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl (4.7 kB)
```

```
from langchain import PromptTemplate
demo_template= 'I want you to act as a text summarizer, summarize these series of texts {text_statement}.'

prompt1 = PromptTemplate(
    input_variables =['text_statement'],
    template = demo_template
    )

prompt1.format(text_statement ='I am a staff of Octave Analytics')
```

```
'I want you to act as a text summarizer, summarize these series of texts I am a staff o
f Octave Analytics.'
```

```
from langchain.llms import OpenAI
from langchain.chains import LLMChain

llms = OpenAI(temperature=0.8)
chain1 = LLMChain(llm=llms,prompt=prompt1)
```

```
/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:117: LangChainDeprecationWarning: The class `langchain_commun
  warn_deprecated(
```

```
# summarizing the prompt text given
chain1.run('I have been a staff of Octave Analytics since July 2023')
```

```
/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:117: LangChai
  warn_deprecated(
'\n\nI have been an employee at Octave Analytics since July 2023. '
```

```
chain1.run('In my previous role, I was responsible for building weekly dashboards and reports for weekly sales and using sales data to forec
```

```
'\n\nThe speaker's previous role involved building dashboards and reports and using sal
es data to forecast and provide financial advice. They currently work with a consulting
company, where they perform exploratory data analysis for small and medium size compani
es to help improve their businesses through operational data.'
```

```
demo_template='''I want you to act as a text translator, translate these series of texts {Statement} to {language}.'''

prompt2 = PromptTemplate(
    input_variables =['statement','language'],
    template = demo_template
    )

prompt2.format(Statement ="'I am a staff of Octave Analytics'",language = 'pidgin english')
```

```
'I want you to act as a text translator, translate these series of texts 'I am a staff
of Octave Analytics' to pidgin english.'
```

```
chain2=LLMChain(llm=llms,prompt=prompt2)

chain2({'Statement':"Hello How are you",'language':'pidgin english'})
```

```
/usr/local/lib/python3.10/dist-packages/langchain_core/_api/deprecation.py:117: LangChainDeprecationWarning: The function `__call__` was
  warn_deprecated(
{'Statement': 'Hello How are you',
 'language': 'pidgin english',
 'text': '\n\n"Abeg, how body dey?"'}
```

```
chain2({'Statement':"I am Jeremiah and I am a staff of Octave Analytics",'language':'pidgin english'})
```

```
{'Statement': 'I am Jeremiah and I am a staff of Octave Analytics',
 'language': 'pidgin english',
 'text': '\n\nI be Jeremiah and I be one staff of Octave Analytics.'}
```

```
chain2({'Statement':"I am Jeremiah and I am a staff of Octave Analytics",'language':'french'})
```

```
{'Statement': 'I am Jeremiah and I am a staff of Octave Analytics',
 'language': 'french',
 'text': "\n\nJe suis Jeremiah et je suis un membre du personnel d'Octave Analytics."}
```

```
pip install pypdf
```

```
Collecting pypdf
  Downloading pypdf-4.1.0-py3-none-any.whl (286 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 286.1/286.1 kB 2.6 MB/s eta 0:00:00
Installing collected packages: pypdf
Successfully installed pypdf-4.1.0
```

```
from langchain.document_loaders import PyPDFLoader
```

```
loader = PyPDFLoader('Feature scaling notes.pdf')
```

```
pages = loader.load()
```

```
len(pages)
```

```
4
```

```
page = pages[1]
```

```
page
```

```
Document(page_content=' \n Feature Scaling  \n \nFeature Scaling is a data preprocessing technique. By preprocessing, we mean the
\ntransformations that are applied to the data before it is fed into some algorithm for some \nprocessing.  \n \nWhat is Feature
Scaling?  \nFeature Scaling is a technique where we stand ardize the range of  all independent features of a \ndata-set. It is also
called Normalization.  \n \nGenerally, when we get raw data, all the features values varies on different scales. It is important \nto
bring all the feature values on the scale so that value of one feature should not dominate over \nthe others and hinder the performance
of  the learning algorithm. This process of bringing all the \nfeatures values on the same scale is called feature scaling. Ensuring
standardised feature values \nimplicitly weights all features equally in their representation.  \n \nFeature scaling or re -scaling of
the features is performed such that they have the properties of \nnormal distribution (most of the time)  where values have standard
deviation = 1, and mean = 0.  \n \n \nWhy Feature Sc aling is required?  \nMost of the real world applied machine learning algorithms
are classification algorithms. Many \nof the classification algorithms works by calculating the distance between data points in space.
If \none feature has a wide range of values, th en this feature is likely to dominate the distance \nmeasure between the data points
over other features. Above this, if this feature proves to be \ninsignificant in the end then, it will be hindering the algorithm
results to a large extent. This will \nresult i n decrease in accuracy of the algorithm.  \n \nFor example :  Let us look at the subset
of wine dataset:  ', metadata={'source': 'Feature scaling notes.pdf', 'page': 1})
```

```
print(page.page_content)
```

```
 Feature Scaling

Feature Scaling is a data preprocessing technique. By preprocessing, we mean the
transformations that are applied to the data before it is fed into some algorithm for some
processing.

What is Feature Scaling?
Feature Scaling is a technique where we stand ardize the range of  all independent features of a
data-set. It is also called Normalization.

Generally, when we get raw data, all the features values varies on different scales. It is important
to bring all the feature values on the scale so that value of one feature should not dominate over
the others and hinder the performance of  the learning algorithm. This process of bringing all the
features values on the same scale is called feature scaling. Ensuring standardised feature values
implicitly weights all features equally in their representation.

Feature scaling or re -scaling of the features is performed such that they have the properties of
normal distribution (most of the time)  where values have standard deviation = 1, and mean = 0.


Why Feature Sc aling is required?
Most of the real world applied machine learning algorithms are classification algorithms. Many
of the classification algorithms works by calculating the distance between data points in space. If
one feature has a wide range of values, th en this feature is likely to dominate the distance
measure between the data points over other features. Above this, if this feature proves to be
insignificant in the end then, it will be hindering the algorithm results to a large extent. This will
result i n decrease in accuracy of the algorithm.

For example :  Let us look at the subset of wine dataset:
```

```
feature_scaling = page.page_content
```

```
feature_scaling
```

```
' \n Feature Scaling  \n \nFeature Scaling is a data preprocessing technique. By prepro
cessing, we mean the \ntransformations that are applied to the data before it is fed in
to some algorithm for some \nprocessing.  \n \nWhat is Feature Scaling?  \nFeature Scal
ing is a technique where we stand ardize the range of  all independent features of a \n
data-set. It is also called Normalization.  \n \nGenerally, when we get raw data, all t
he features values varies on different scales. It is important \nto bring all the featu
re values on the scale so that value of one feature should not dominate over \nthe othe
rs and hinder the performance of  the learning algorithm. This process of bringing all
```

```
demo_template='''I want you to act as a text reader, and answer questions from a given text {text} answer this {question}.'''

prompt3 = PromptTemplate(
    input_variables =['text','question'],
    template = demo_template
    )

prompt3.format(text ="'I am a staff of Octave Analytics'", question = 'who are you?')
```

```
    'I want you to act as a text reader, and answer questions from a given text 'I am a sta
    ff of Octave Analytics' answer this who are you? '
```

```
chain3=LLMChain(llm=llms,prompt=prompt3)
```

```
chain3({'text':feature_scaling,'question':"what is feature scaling"})
```

```
    {'text': "\n\nFeature scaling is a data preprocessing technique where the range of all independent features in a dataset is
    standardized or normalized. This is done to ensure that no single feature dominates the algorithm's calculations and to improve the
    overall performance and accuracy of the algorithm. ",
     'question': 'what is feature scaling'}
```

```
chain3({'text':feature_scaling,'question':"what are the properties of feature scaling"})
```

```
    {'text': '\nThe properties of feature scaling include bringing all independent features onto the same scale, ensuring equal
    representation of all features, and having a normal distribution with standard deviation of 1 and mean of 0. Feature scaling is
    required to prevent one feature from dominating the distance measure between data points and hindering the performance of the
    algorithm. This helps to improve the accuracy of classification algorithms that rely on distance calculations. ',
     'question': 'what are the properties of feature scaling'}
```

```
page2 = pages[2]
```

```
print(page2.page_content)
```

```
    The range of value for “Magnesium” feature is 80 - 100 or above, while range of values of
    “Malic_Acid”  feature is ranging 1.something.
    Now if we apply  distance formula(as we do in case of KNN) on two data points here,
    Magnesium will dominate the distance value to a greater extent than any of the other feature,
    thus resulting in wrong predictions later.

    Note - Distance Formula is given as d = (( x1 - x2 )^2 + (y1 -y2)^2....)^½

    Let’s understand this better with the help of an other  example –
    Suppose you have a company employees’ data. Now the age of employees in a company may be
    between 21 -70 years, the size of the house they live is 500 -5000 Sq feet and t heir salaries may
    range from 30000–80000. In this situation if you use a simple Euclidean metric, the age feature
    will not play any role because it is several order smaller than other features.  However, it may
    contain some important information that may be  useful for the task. Here, you may want to
    normalize the features independently to the same scale, say [0,1], so they contribute equally
    while computing the distance.”

    How Feature Scaling is applied in sklearn?

     There are many ways by which we can apply  feature scaling on the dataset:
    1. The easiest way of scaling is to use - preprocessing.scale() function
    A numpy array of values is given as input and output is numpy array with scaled values.

    This will scale the values in such a way that mean of the values  will be 0 and standard
```

```
feature_scale2 = page.page_content
```

```
chain3({'text':feature_scale2,'question':"using the distance formula calculate d when y1=3, y2=2, x1=4, x2=3"})
```

```
    {'text': ' \n\nThe distance, d, when y1=3, y2=2, x1=4, x2=3 would be calculated as:\nd = sqrt((3-2)^2 + (4-3)^2)\n= sqrt(1 + 1)\n=
    sqrt(2)\n= 1.4142\n\nWithout feature scaling, if one feature has a larger range of values, it can dominate the distance calculation and
    potentially give inaccurate results. By standardizing the features, all features will have the same scale and will not affect the
    distance calculation significantly. This is why feature scaling is required for many machine learning algorithms.',
     'question': 'using the distance formula calculate d when y1=3, y2=2, x1=4, x2=3'}
```

```
chain3({'text':feature_scale2,'question':"how do we implement feature scaling"})
```

```
    {'text': '\n\nOne way to implement feature scaling is by using techniques such as standardization or normalization. Standardization
    involves transforming the data such that it has a mean of 0 and a standard deviation of 1. This can be done by subtracting the mean
    from each data point and then dividing by the standard deviation. Normalization involves transforming the data such that it falls
    within a specific range, typically between 0 and 1. This can be done by subtracting the minimum value from each data point and then
    dividing by the difference between the maximum and minimum values. Other techniques such as log transformation or z-score normalization
    can also be used for feature scaling. The exact method used will depend on the data and the specific algorithm being used. ',
     'question': 'how do we implement feature scaling'}
```

```
chain3({'text':feature_scale2,'question':"which libraries can i use for feature scaling"})
```

```
    {'text': '\n\nYou can use sklearn.preprocessing library or pandas library in Python for feature scaling.',
     'question': 'which libraries can i use for feature scaling'}
```