```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
#import tensorflow as tf
#tf.test.gpu_device_name()
```

```
pip install python-docx
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting python-docx
  Downloading python-docx-0.8.11.tar.gz (5.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.6/5.6 MB 46.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: lxml>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from python-docx) (4.9.2)
Building wheels for collected packages: python-docx
  Building wheel for python-docx (setup.py) ... done
  Created wheel for python-docx: filename=python_docx-0.8.11-py3-none-any.whl size=184491 sha256=0eef7ca87e5a3a390b08e2373071f52dab
  Stored in directory: /root/.cache/pip/wheels/80/27/06/837436d4c3bd989b957a91679966f207bfd71d358d63a8194d
Successfully built python-docx
Installing collected packages: python-docx
Successfully installed python-docx-0.8.11
```

```
import sklearn
```

Data Extraction from the given documents

```
import pandas as pd
import re
from docx import Document

def extract_clauses(document):
    # Define the regular expression pattern to identify clauses
    pattern = r'[^.!?]+[.!?]'

    # Extract clauses from the document using the regular expression pattern
    clauses = re.findall(pattern, document)

    return clauses

# List of document file names
document_files = ['/content/drive/MyDrive/SampleDocs/20201023_GG_Loan Agreement.docx', '/content/drive/MyDrive/SampleDocs/20201028_MyTTe
'/content/drive/MyDrive/SampleDocs/Consulting Agreement- Nikhil D.docx','/content/drive/MyDrive/SampleDocs/Demo Joint Venture Agreement
'/content/drive/MyDrive/SampleDocs/FOUNDERS AGREEMENT-December 03 2017 (Final Version) (for compare).docx','/content/drive/MyDrive/Sampl
'/content/drive/MyDrive/SampleDocs/Sale Agreement 14.9.docx','/content/drive/MyDrive/SampleDocs/Tea Cozie Vendor Agreement 250919.docx',

# List to store the extracted clauses
extracted_clauses = []

# Iterate over the document files
for file in document_files:
    doc = Document(file)
    document_text = ''

    # Extract the text from the document and remove underlines
    for paragraph in doc.paragraphs:
        text = paragraph.text

        # Remove underlines
        text = re.sub(r'_', '', text)

        # Remove integers
        text = re.sub(r'\d', '', text)

        document_text += text + ' '

    clauses = extract_clauses(document_text)
    extracted_clauses.extend(clauses)

# Create a DataFrame with the extracted clauses
data = pd.DataFrame({'Clause': extracted_clauses})

# Save the DataFrame to a CSV file
```

```
data.to_csv('clauses_dataset.csv', index=False)
```

```
pd.read_csv('clauses_dataset.csv')
```

	Clause
0	LOAN AGREEMENT This Loan Agreement (“Agreement...
1	UMPPTC, and its registered office situated at...
2	, Paras Majestic, Near Aura Mall, Trilanga Co...
3	(“Business”).
4	The Lender has agreed to tender loan of prin...
...	...
3102	) BILLING FOR UTILITIES THAT REMAIN IN LANDL...
3103	If the charges are more than the amount paid ...
3104	b) If Tenant has been late on any month’s ren...
3105	) FURNACE UPKEEP AND MAINTENANCE: Tenant(s) a...
3106	) SMOKING: No smoking will be allowed in the...

3107 rows × 1 columns

Data Preproccesing

```
import re
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

def preprocess_text(text):
    # Convert text to lowercase
    text = text.lower()

    # Remove numbers
    text = re.sub(r'\d+_:', '', text)

    # Remove special characters and punctuation
    text = re.sub(r'[^\w\s]', '', text)

    # Tokenize the text
    tokens = word_tokenize(text)

    # Remove stop words
    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]

    # Lemmatize the tokens
    lemmatizer = WordNetLemmatizer()
    tokens = [lemmatizer.lemmatize(token) for token in tokens]

    # Join the tokens back into a single string
    preprocessed_text = ' '.join(tokens)

    return preprocessed_text

# Load the CSV dataset
data = pd.read_csv('clauses_dataset.csv')

# Preprocess the text data
data['PreprocessedText'] = data['Clause'].apply(preprocess_text)

# Save the preprocessed data to a new CSV file
data.to_csv('preprocessed_dataset.csv', index=False)
```

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

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

```
pd.read_csv('preprocessed_dataset.csv')
```

	Clause	PreprocessedText
0	LOAN AGREEMENT This Loan Agreement (“Agreement...	loan agreement loan agreement agreement entere...
1	UMPPTC, and its registered office situated at...	umpptc registered office situated house
2	, Paras Majestic, Near Aura Mall, Trilanga Co...	para majestic near aura mall trilanga colony b...
3	(“Business”).	business
4	The Lender has agreed to tender loan of prin...	lender agreed tender loan principal sum inr ru...
...	...	...
3102	) BILLING FOR UTILITIES THAT REMAIN IN LANDL...	billing utility remain landlord name landlord ...
3103	If the charges are more than the amount paid ...	charge amount paid tenant month tenant pay dif...
3104	b) If Tenant has been late on any month’s ren...	b tenant late month rent year owes additional ...
3105	) FURNACE UPKEEP AND MAINTENANCE: Tenant(s) a...	furnace upkeep maintenance tenant agree clean ...
3106	) SMOKING: No smoking will be allowed in the...	smoking smoking allowed unit

3107 rows × 2 columns

Feature Extracting Process

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer

# Load the preprocessed dataset
data = pd.read_csv('preprocessed_dataset.csv')

# Fill NaN values with an empty string
data['PreprocessedText'].fillna('', inplace=True)

# Extract features using Bag-of-Words (BoW)
bow_vectorizer = CountVectorizer()
bow_features = bow_vectorizer.fit_transform(data['PreprocessedText'])
bow_feature_names = bow_vectorizer.get_feature_names_out()

# Extract features using TF-IDF
tfidf_vectorizer = TfidfVectorizer()
tfidf_features = tfidf_vectorizer.fit_transform(data['PreprocessedText'])
tfidf_feature_names = tfidf_vectorizer.get_feature_names_out()

# Convert the features to DataFrames for further analysis or merging
bow_df = pd.DataFrame(bow_features.toarray(), columns=bow_feature_names)
tfidf_df = pd.DataFrame(tfidf_features.toarray(), columns=tfidf_feature_names)

# Optionally, you can merge the feature DataFrames with the original dataset
merged_data = pd.concat([data, bow_df, tfidf_df], axis=1)

# Save the merged data to a new CSV file
merged_data.to_csv('feature_dataset.csv', index=False)
```

```
data_1 = pd.read_csv('feature_dataset.csv')
data_1
```

	Clause	PreprocessedText	aaa	aadhaar	aatm	ab	abac	abandon	abando
0	LOAN AGREEMENT This Loan Agreement (“Agreement...	loan agreement loan agreement agreement entere...	0	0	0	0	0	0	
1	UMPPTC, and its registered office situated at...	umpptc registered office situated house	0	0	0	0	0	0	
2	, Paras Majestic, Near Aura Mall, Trilanga Co...	para majestic near aura mall trilanga colony b...	0	2	0	0	0	0	
3	(“Business”).	business	0	0	0	0	0	0	
4	The Lender has agreed to tender loan of prin...	lender agreed tender loan principal sum inr ru...	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	
3102	) BILLING FOR UTILITIES THAT REMAIN IN LANDL...	billing utility remain landlord name landlord ...	0	0	0	0	0	0	
3103	If the charges are more than the amount paid ...	charge amount paid tenant month tenant pay dif...	0	0	0	0	0	0	

```
description = data_1.describe()
print(description)
```

```
                aaa      aadhaar         aatm           ab         abac  \
count  3107.000000  3107.000000  3107.000000  3107.000000  3107.000000
mean      0.000322     0.000644     0.000322     0.000322     0.000966
std       0.017940     0.035881     0.017940     0.017940     0.031063
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       1.000000     2.000000     1.000000     1.000000     1.000000

            abandon    abandoned  abandonment          abc        abide  ...  \
count  3107.000000  3107.000000  3107.000000  3107.000000  3107.000000  ...
mean      0.000322     0.001287     0.000322     0.000644     0.002575  ...
std       0.017940     0.035863     0.017940     0.025367     0.056683  ...
min       0.000000     0.000000     0.000000     0.000000     0.000000  ...
25%       0.000000     0.000000     0.000000     0.000000     0.000000  ...
50%       0.000000     0.000000     0.000000     0.000000     0.000000  ...
75%       0.000000     0.000000     0.000000     0.000000     0.000000  ...
max       1.000000     1.000000     1.000000     1.000000     2.000000  ...

           yield.1      youll.1      young.1      youre.1     zivame.1  \
count  3107.000000  3107.000000  3107.000000  3107.000000  3107.000000
mean      0.000087     0.000152     0.000075     0.000221     0.000068
std       0.004823     0.008454     0.004163     0.009132     0.003818
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       0.268836     0.471220     0.232066     0.450698     0.212824

              zl.1       zone.1       zoom.1     zoraya.1         ép.1
count  3107.000000  3107.000000  3107.000000  3107.000000  3107.000000
mean      0.000090     0.000100     0.000068     0.000431     0.000109
std       0.004999     0.005548     0.003818     0.011312     0.006066
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       0.278633     0.309268     0.212824     0.431145     0.338139

[8 rows x 8878 columns]
```

```
from sklearn.feature_selection import SelectKBest, f_classif

# Read the features dataset
data = pd.read_csv('preprocessed_dataset.csv')

X = data.drop('Clause', axis=1)
y = data['Clause']
```

```
# Perform feature selection
k = 10  # Select the top k features
selector = SelectKBest(score_func=f_classif, k=k)
X_selected = selector.fit_transform(X, y)

# Get the selected feature names
selected_feature_names = X.columns[selector.get_support()]

# Print the selected features
print(selected_feature_names)
```

✓ 19s completed at 4:34 PM