Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJ19ITL504 **DATE:6/12/22**

COURSE NAME: Artificial Intelligence Laboratory **CLASS: TYBTech-IT**

EXPERIMENT NO. 7

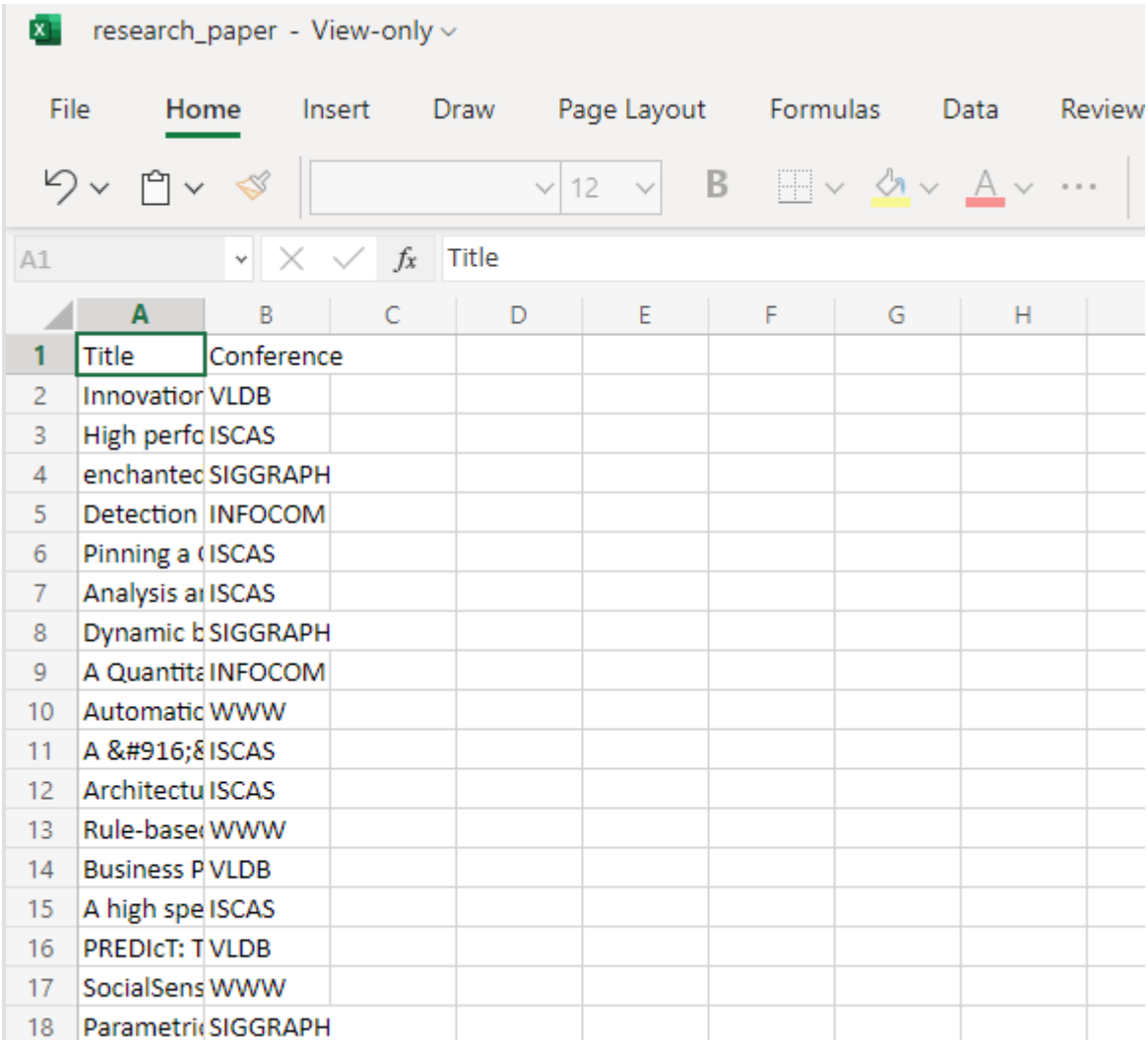
CO/LO: Apply NLP algorithms and methods to solve domain-specific problems

AIM: To perform Text Classification using Spacy.

DESCRIPTIONOF EXPERIMENT:

Spacy is a popular and easy-to-use natural language processing library in Python. It provides current state-of-the-art accuracy and speed levels and has an active open-source community. However, since SpaCy is a relatively new NLP library, it's not as widely adopted as NLTK.

DATASET :



	A	B	C	D	E	F	G	H
1	Title	Conference						
2	Innovatior	VLDB						
3	High perfo	ISCAS						
4	enchantec	SIGGRAPH						
5	Detection	INFOCOM						
6	Pinning a (	ISCAS						
7	Analysis a	ISCAS						
8	Dynamic b	SIGGRAPH						
9	A Quantita	INFOCOM						
10	Automatic	WWW						
11	A Δ&	ISCAS						
12	Architectu	ISCAS						
13	Rule-base	WWW						
14	Business P	VLDB						
15	A high spe	ISCAS						
16	PREDICT: T	VLDB						
17	SocialSens	WWW						
18	Parametri	SIGGRAPH						

TECHNOLOGY STACK USED:Python (Google colab)

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

CODE&OUTPUT:

```
import pandas as pd
import numpy as np # linear algebra
import seaborn as sns
import matplotlib.pyplot as plt
import base64
import string
import re
import nltk
nltk.download('stopwords')

from collections import Counter
from nltk.corpus import stopwords
stopwords = stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
df = pd.read_csv('research_paper.csv')
df.head()
```

	Title	Conference
0	Innovation in Database Management: Computer Sc...	VLDB
1	High performance prime field multiplication fo...	ISCAS
2	enchanted scissors: a scissor interface for su...	SIGGRAPH
3	Detection of channel degradation attack by Int...	INFOCOM
4	Pinning a Complex Network through the Betweenn...	ISCAS

```
[ ] df.shape
```

```
(2507, 2)
```

```
[ ] df.isnull().sum()
```

```
Title         0
Conference    0
dtype: int64
```

```
[ ] df['Conference'].nunique()
```

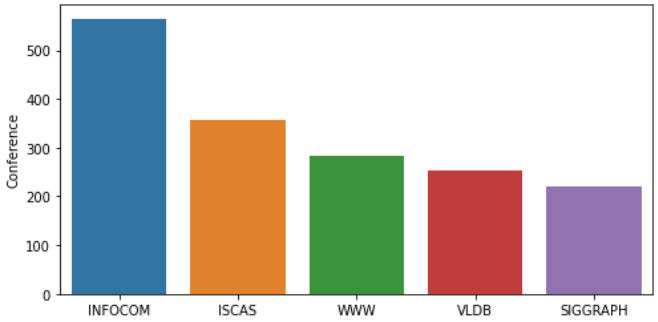
```
5
```

```
from sklearn.model_selection import train_test_split
train, test = train_test_split(df, test_size=0.33, random_state=42)
```

```
[ ] print('Research title sample:', train['Title'].iloc[0])
    print('Conference of this paper:', train['Conference'].iloc[0])
    print('Training Data Shape:', train.shape)
    print('Testing Data Shape:', test.shape)
```

```
Research title sample: Cooperating with Smartness: Using Heterogeneous Smart Antennas in Ad-Hoc Networks.
Conference of this paper: INFOCOM
Training Data Shape: (1679, 2)
Testing Data Shape: (828, 2)
```

```
[ ]  fig = plt.figure(figsize=(8,4))
     sns.barplot(x = train['Conference'].unique(), y=train['Conference'].value_counts())
     plt.show()
```

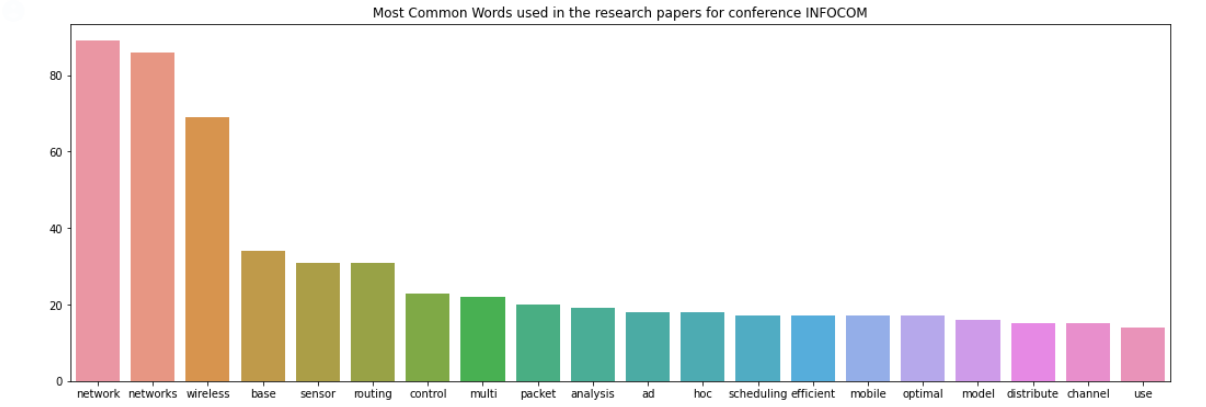



```
[ ]  import spacy

     nlp = spacy.load('en_core_web_sm')
     punctuations = string.punctuation

     # Define function to cleanup text by removing personal pronouns, stopwords, and puncuation
     def cleanup_text(docs, logging=False):
         texts = []
         counter = 1
         for doc in docs:
             if counter % 1000 == 0 and logging:
                 print("Processed %d out of %d documents." % (counter, len(docs)))
             counter += 1
             doc = nlp(doc, disable=['parser', 'ner'])
             tokens = [tok.lemma_.lower().strip() for tok in doc if tok.lemma_ != '-PRON-']
             tokens = [tok for tok in tokens if tok not in stopwords and tok not in punctuations]
             tokens = ' '.join(tokens)
             texts.append(tokens)
         return pd.Series(texts)
```

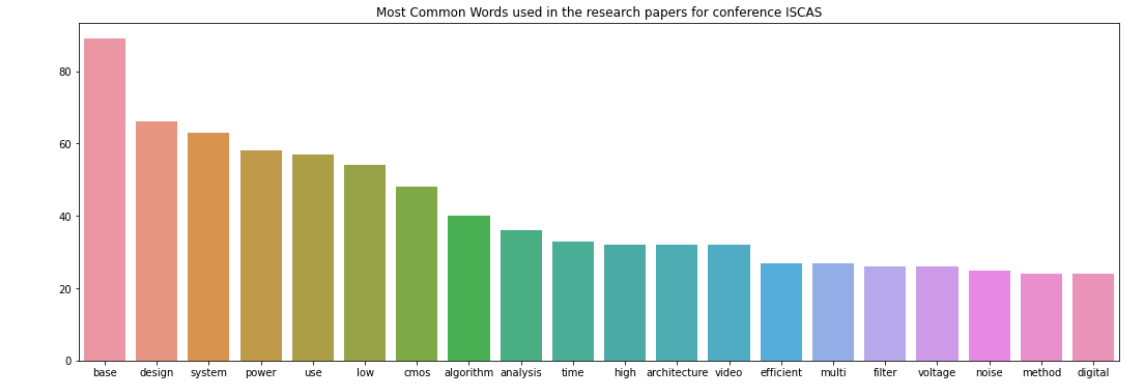
```
[ ]  INFO_text = [text for text in train[train['Conference'] == 'INFOCOM']['Title']]

     IS_text = [text for text in train[train['Conference'] == 'ISCAS']['Title']]

     INFO_clean = cleanup_text(INFO_text)
     INFO_clean = ' '.join(INFO_clean).split()

     IS_clean = cleanup_text(IS_text)
     IS_clean = ' '.join(IS_clean).split()

     INFO_counts = Counter(INFO_clean)
     IS_counts = Counter(IS_clean)
```

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

```
[ ] INFO_common_words = [word[0] for word in INFO_counts.most_common(20)]
    INFO_common_counts = [word[1] for word in INFO_counts.most_common(20)]

    fig = plt.figure(figsize=(18,6))
    sns.barplot(x=INFO_common_words, y=INFO_common_counts)
    plt.title('Most Common Words used in the research papers for conference INFOCOM')
    plt.show()
```




```
[ ] IS_common_words = [word[0] for word in IS_counts.most_common(20)]
    IS_common_counts = [word[1] for word in IS_counts.most_common(20)]

    fig = plt.figure(figsize=(18,6))
    sns.barplot(x=IS_common_words, y=IS_common_counts)
    plt.title('Most Common Words used in the research papers for conference ISCAS')
    plt.show()
```



Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

```
test.head()
```

	Title	Conference
2121	Architectural Issues in Distributed Data Base ...	VLDB
56	Compressive sampling of EMG bio-signals.	ISCAS
2479	User-Centered Modeling of Interactive Web Sites.	WWW
1292	A Decomposition Method for Transmission Schedu...	INFOCOM
1599	ASCENT: Adaptive Self-Configuring sEnsor Netwo...	INFOCOM

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.base import TransformerMixin
from sklearn.pipeline import Pipeline
from sklearn.svm import LinearSVC
from sklearn.feature_extraction._stop_words import ENGLISH_STOP_WORDS
from sklearn.metrics import accuracy_score
from nltk.corpus import stopwords
import string
import re
import spacy
spacy.load('en_core_web_sm')
from spacy.lang.en import English
parser = English()
```

```
STOPLIST = set(stopwords.words('english') + list(ENGLISH_STOP_WORDS))
SYMBOLS = " ".join(string.punctuation).split(" ") + ["-", "...", "”", "”"]
```

```
class CleanTextTransformer(TransformerMixin):
    def transform(self, X, **transform_params):
        return [cleanText(text) for text in X]
    def fit(self, X, y=None, **fit_params):
        return self
    def get_params(self, deep=True):
        return {}

def cleanText(text):
    text = text.strip().replace("\n", " ").replace("\r", " ")
    text = text.lower()
    return text

def tokenizeText(sample):
    tokens = parser(sample)
    lemmas = []
    for tok in tokens:
        lemmas.append(tok.lemma_.lower().strip() if tok.lemma_ != "-PRON-" else tok.lower_)
    tokens = lemmas
    tokens = [tok for tok in tokens if tok not in STOPLIST]
    tokens = [tok for tok in tokens if tok not in SYMBOLS]
    return tokens
```

7s completed at 20:43

```
[ ] def printNMostInformative(vectorizer, clf, N):
        feature_names = vectorizer.get_feature_names()
        coefs_with_fns = sorted(zip(clf.coef_[0], feature_names))
        topClass1 = coefs_with_fns[:N]
        topClass2 = coefs_with_fns[:-(N + 1):-1]
        print("Class 1 best: ")
        for feat in topClass1:
            print(feat)
        print("Class 2 best: ")
        for feat in topClass2:
            print(feat)

    vectorizer = CountVectorizer(tokenizer=tokenizeText, ngram_range=(1,1))
    clf = LinearSVC()
    pipe = Pipeline([('cleanText', CleanTextTransformer()), ('vectorizer', vectorizer), ('clf', clf)])
```

```
[ ] # data
    train1 = train['Title'].tolist()
    labelsTrain1 = train['Conference'].tolist()

    test1 = test['Title'].tolist()
    labelsTest1 = test['Conference'].tolist()
    # train
    pipe.fit(train1, labelsTrain1)

    # test
    preds = pipe.predict(test1)
    print("accuracy:", accuracy_score(labelsTest1, preds))
    print("Top 10 features used to predict: ")

    printNMostInformative(vectorizer, clf, 10)
```

```
pipe = Pipeline([('cleanText', CleanTextTransformer()), ('vectorizer', vectorizer)])

transform = pipe.fit_transform(train1, labelsTrain1)

vocab = vectorizer.get_feature_names()


for i in range(len(train1)):

    s = ""

    indexIntoVocab = transform.indices[transform.indptr[i]:transform.indptr[i+1]]

    numOccurences = transform.data[transform.indptr[i]:transform.indptr[i+1]]

    for idx, num in zip(indexIntoVocab, numOccurences):

        s += str((vocab[idx], num))
```

```
accuracy: 0.37922705314009664
Top 10 features used to predict:
Class 1 best:
(0.004291049300077565, '')
Class 2 best:
(0.004291049300077565, '')
/usr/local/lib/python3.7/dist-packages/sklearn/svm/_base.py:1208: ConvergenceWarning: Liblinear failed to converge, increase the number of iterations.
  ConvergenceWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated i
  warnings.warn(msg, category=FutureWarning)
```

```
[ ] from sklearn import metrics
    print(metrics.classification_report(labelsTest1, preds,
                                        target_names=df['Conference'].unique()))
```

	precision	recall	f1-score	support
VLDB	0.00	0.00	0.00	159
ISCAS	0.43	0.89	0.58	299
SIGGRAPH	0.23	0.45	0.31	106
INFOCOM	0.00	0.00	0.00	139
WWW	0.00	0.00	0.00	125
accuracy			0.38	828
macro avg	0.13	0.27	0.18	828
weighted avg	0.18	0.38	0.25	828

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
  _warn_prf(average, modifier, msg_start, len(result))
```

CONCLUSION:

Hence,We have implemented text classification with the help of SpaCy.

REFERENCES:

[1] Machine Learning for Text Classification Using SpaCy in Python from https://towardsdatascience.com/machine-learning-for-text-classification-using-spacy-in-python-b276b4051a49

[2] Tutorial: Text Classification Using spaCy from https://www.kaggle.com/code/satishgunjal/tutorial-text-classification-using-spacy