CSE508
(Information Retrieval)
Assignment 2
Report

Abhey Kalia - 2020420
Bhavya Jain - 2020428
Utkarsh Arora - 2020143

Q1)

As all preprocessing was same in A1, hence using the same screenshots.

Data Preprocessing

(i) Retrieved the data present inside the <TEXT> … </TEXT> and <TITLE> … </TITLE> tags using file reader from 'os' library and concatenated the strings obtained using a single space. We checked for <text> tag by word.lower() and appended until </text> was not present and did the same for <title> tag.

Before processing:

1. **cranfield0036:**

```
<DOC>
<DOCNO>
36
</DOCNO>
<TITLE>
supersonic flow around blunt bodies .
</TITLE>
<AUTHOR>
serbin,h.
</AUTHOR>
<BIBLIO>
j. ae. scs. 25, 1958, 58.
</BIBLIO>
<TEXT>
the newtonian theory of impact has been shown to be
useful for pressure calculations on the forward facing part
of bodies moving at high speed .  it is now a familiar practice
to use this information to calculate nonviscous velocities at the
wall and then to estimate rates of heat transfer .  this procedure
is perhaps open to question,. heat-transfer rates depend
on velocity gradients which are not given by the newtonian
analysis .  nor can one obtain information on boundary-layer
stability or all the body stability derivatives .  it seems, therefore,
inevitable that, as design proceeds with these hypersonic
missiles, there will be a greater need for more accurate aerodynamic
theories either to predict what will happen in unfamiliar
flight conditions or to effect an extrapolation from a known test
result to the design condition .
</TEXT>
</DOC>
```

2. **cranfield0041:**

```
<DOC>
<DOCNO>
41
</DOCNO>
<TITLE>
on transition experiments at moderate supersonic speeds .
</TITLE>
<AUTHOR>
morkovin,m.v.
</AUTHOR>
<BIBLIO>
j. ae. scs. 24, 1957, 480.
</BIBLIO>
<TEXT>
  studies of transition over a flat plate at mach number 1.76
were carried out using a hot-wire anemometer as one of the
principal tools .  the nature and measurements of free-stream
disturbances at supersonic speeds are analyzed .  the experimental
results are interpreted in the light of present overall information
on transition at supersonic speeds and conclusions as
to further fruitful experiments are drawn .
</TEXT>
</DOC>
```

3. **cranfield0067:**

```
<DOC>
<DOCNO>
67
</DOCNO>
<TITLE>
dynamic stability of vehicles traversing ascending
or descending paths through the atmosphere .
</TITLE>
<AUTHOR>
tobak and allen.
</AUTHOR>
<BIBLIO>
naca tn.4275, 1958.
</BIBLIO>
<TEXT>
  an analysis is given of the oscillatory motions of vehicles which
traverse ascending and descending paths through the atmosphere at high
speed .  the specific case of a skip path is examined in detail, and
this leads to a form of solution for the oscillatory motion which should
recur over any trajectory .  the distinguishing feature of this form is
the appearance of the bessel rather than the trigonometric function as
the characteristic mode of oscillation .
</TEXT>
</DOC>
```

4. **cranfield0071:**

```
<DOC>
<DOCNO>
71
</DOCNO>
<TITLE>
laminar boundary layer behind shock advancing into
stationary fluid .
</TITLE>
<AUTHOR>
mirels,h.
</AUTHOR>
<BIBLIO>
naca tn.3401, 1955.
</BIBLIO>
<TEXT>
  a study was made of the laminar compressible boundary layer induced
by a shock wave advancing into a stationary fluid bounded by a wall .
for weak shock waves, the boundary layer is identical with that which
occurs when an infinite wall is impulsively set into uniform motion
shocks .
  velocity and temperature profiles, recovery factors, and skinfriction
and heat-transfer coefficients are tabulated for a wide range
of shock strengths .
</TEXT>
</DOC>
```

5. **cranfield0098:**

```
<DOC>
<DOCNO>
98
</DOCNO>
<TITLE>
heat transfer by laminar flow to a rotating plate .
</TITLE>
<AUTHOR>
millsaps,k. and pohlhausen,k.
</AUTHOR>
<BIBLIO>
j. ae. scs. 19, 1952, 120.
</BIBLIO>
<TEXT>
  an exact solution of the heat-transfer problem for the von
karman example of the laminar flow of a viscous fluid over a
rotating plate is given in dimensionless form and physically discussed
.  the solution is explicitly given for a constant temperature
on the plate with viscous dissipation included .  the numerical
results are given for prandtl numbers from 0.5 to 10 .
</TEXT>
</DOC>
```

After processing:

1. **cranfield0036**:

```
supersonic flow around blunt bodies .
 the newtonian theory of impact has been shown to be
 useful for pressure calculations on the forward facing part
 of bodies moving at high speed .  it is now a familiar practice
 to use this information to calculate nonviscous velocities at the
 wall and then to estimate rates of heat transfer .  this procedure
 is perhaps open to question,. heat-transfer rates depend
 on velocity gradients which are not given by the newtonian
 analysis .  nor can one obtain information on boundary-layer
 stability or all the body stability derivatives .  it seems, therefore,
 inevitable that, as design proceeds with these hypersonic
 missiles, there will be a greater need for more accurate aerodynamic
 theories either to predict what will happen in unfamiliar
 flight conditions or to effect an extrapolation from a known test
 result to the design condition .
```

2. **cranfield0041:**

```
on transition experiments at moderate supersonic speeds .
   studies of transition over a flat plate at mach number 1.76
 were carried out using a hot-wire anemometer as one of the
 principal tools .  the nature and measurements of free-stream
 disturbances at supersonic speeds are analyzed .  the experimental
 results are interpreted in the light of present overall information
 on transition at supersonic speeds and conclusions as
 to further fruitful experiments are drawn .
```

3. **cranfield0067:**

```
dynamic stability of vehicles traversing ascending
 or descending paths through the atmosphere .
   an analysis is given of the oscillatory motions of vehicles which
 traverse ascending and descending paths through the atmosphere at high
 speed .  the specific case of a skip path is examined in detail, and
 this leads to a form of solution for the oscillatory motion which should
 recur over any trajectory .  the distinguishing feature of this form is
 the appearance of the bessel rather than the trigonometric function as
 the characteristic mode of oscillation .
```

4. **cranfield0071:**

```
laminar boundary layer behind shock advancing into
 stationary fluid .
   a study was made of the laminar compressible boundary layer induced
 by a shock wave advancing into a stationary fluid bounded by a wall .
 for weak shock waves, the boundary layer is identical with that which
 occurs when an infinite wall is impulsively set into uniform motion
 shocks .
   velocity and temperature profiles, recovery factors, and skinfriction
 and heat-transfer coefficients are tabulated for a wide range
 of shock strengths .
```

5. **cranfield0098:**

```
heat transfer by laminar flow to a rotating plate .
   an exact solution of the heat-transfer problem for the von
karman example of the laminar flow of a viscous fluid over a
rotating plate is given in dimensionless form and physically discussed
.  the solution is explicitly given for a constant temperature
on the plate with viscous dissipation included .  the numerical
results are given for prandtl numbers from 0.5 to 10 .
```

(ii)

1. **For lowercasing the words:**
   We used python's lower() method on every word that was present in the file and wrote the resultant words onto the same file.
2. For tokenization:
   We used nltk library and used it's word_tokenize() method to tokenize the text present in the file.
3. For removing stopwords:
   Imported nltk's corpus of stop words and made a set of these stopwords, now we compared every word present in the file and if it was a stopword we did not write it and it wasn't we wrote the word back into the file. Since the read operation is done sentence wise from left to right, the position of the remaining words was unchanged.
4. Removing punctuations:
   We imported string class and stripped down any punctuations and appended the words if there is a punctuation between them else the space was maintained.
   E.g.

   ain't → aint

   right-handed → righthanded

   This ain't it? right. → This aint it right
5. Removing blank spaces:
   This was implicitly handled by the word_tokenize() method of the nltk library, since the method only tokenizes those strings which have a non-empty character present.

1. cranfield0036

before lower case:

```
['supersonic', 'flow', 'around', 'blunt', 'bodies', '.', 'the',
'newtonian', 'theory', 'of', 'impact', 'has', 'been', 'shown',
'to', 'be', 'useful', 'for', 'pressure', 'calculations', 'on',
'the', 'forward', 'facing', 'part', 'of', 'bodies', 'moving',
'at', 'high', 'speed', '.', 'it', 'is', 'now', 'a', 'familiar',
'practice', 'to', 'use', 'this', 'information', 'to',
```

'calculate', 'nonviscous', 'velocities', 'at', 'the', 'wall',
'and', 'then', 'to', 'estimate', 'rates', 'of', 'heat',
'transfer', '.', 'this', 'procedure', 'is', 'perhaps', 'open',
'to', 'question', ',', '.', 'heat-transfer', 'rates', 'depend',
'on', 'velocity', 'gradients', 'which', 'are', 'not', 'given',
'by', 'the', 'newtonian', 'analysis', '.', 'nor', 'can', 'one',
'obtain', 'information', 'on', 'boundary-layer', 'stability',
'or', 'all', 'the', 'body', 'stability', 'derivatives', '.',
'it', 'seems', ',', 'therefore', ',', 'inevitable', 'that',
',', 'as', 'design', 'proceeds', 'with', 'these', 'hypersonic',
'missiles', ',', 'there', 'will', 'be', 'a', 'greater', 'need',
'for', 'more', 'accurate', 'aerodynamic', 'theories', 'either',
'to', 'predict', 'what', 'will', 'happen', 'in', 'unfamiliar',
'flight', 'conditions', 'or', 'to', 'effect', 'an',
'extrapolation', 'from', 'a', 'known', 'test', 'result', 'to',
'the', 'design', 'condition', '.']

after lowercase:

['supersonic', 'flow', 'around', 'blunt', 'bodies', '.', 'the',
'newtonian', 'theory', 'of', 'impact', 'has', 'been', 'shown',
'to', 'be', 'useful', 'for', 'pressure', 'calculations', 'on',
'the', 'forward', 'facing', 'part', 'of', 'bodies', 'moving',
'at', 'high', 'speed', '.', 'it', 'is', 'now', 'a', 'familiar',
'practice', 'to', 'use', 'this', 'information', 'to',
'calculate', 'nonviscous', 'velocities', 'at', 'the', 'wall',
'and', 'then', 'to', 'estimate', 'rates', 'of', 'heat',
'transfer', '.', 'this', 'procedure', 'is', 'perhaps', 'open',
'to', 'question', ',', '.', 'heat-transfer', 'rates', 'depend',
'on', 'velocity', 'gradients', 'which', 'are', 'not', 'given',
'by', 'the', 'newtonian', 'analysis', '.', 'nor', 'can', 'one',
'obtain', 'information', 'on', 'boundary-layer', 'stability',
'or', 'all', 'the', 'body', 'stability', 'derivatives', '.',
'it', 'seems', ',', 'therefore', ',', 'inevitable', 'that',
',', 'as', 'design', 'proceeds', 'with', 'these', 'hypersonic',
'missiles', ',', 'there', 'will', 'be', 'a', 'greater', 'need',
'for', 'more', 'accurate', 'aerodynamic', 'theories', 'either',
'to', 'predict', 'what', 'will', 'happen', 'in', 'unfamiliar',
'flight', 'conditions', 'or', 'to', 'effect', 'an',
'extrapolation', 'from', 'a', 'known', 'test', 'result', 'to',
'the', 'design', 'condition', '.']

before removal of stopwords:

['supersonic', 'flow', 'around', 'blunt', 'bodies', '.', 'the',
'newtonian', 'theory', 'of', 'impact', 'has', 'been', 'shown',
'to', 'be', 'useful', 'for', 'pressure', 'calculations', 'on',
'the', 'forward', 'facing', 'part', 'of', 'bodies', 'moving',
'at', 'high', 'speed', '.', 'it', 'is', 'now', 'a', 'familiar',
'practice', 'to', 'use', 'this', 'information', 'to',
'calculate', 'nonviscous', 'velocities', 'at', 'the', 'wall',
'and', 'then', 'to', 'estimate', 'rates', 'of', 'heat',
'transfer', '.', 'this', 'procedure', 'is', 'perhaps', 'open',
'to', 'question', ',', '.', 'heat-transfer', 'rates', 'depend',
'on', 'velocity', 'gradients', 'which', 'are', 'not', 'given',
'by', 'the', 'newtonian', 'analysis', '.', 'nor', 'can', 'one',
'obtain', 'information', 'on', 'boundary-layer', 'stability',
'or', 'all', 'the', 'body', 'stability', 'derivatives', '.',
'it', 'seems', ',', 'therefore', ',', 'inevitable', 'that',
',', 'as', 'design', 'proceeds', 'with', 'these', 'hypersonic',
'missiles', ',', 'there', 'will', 'be', 'a', 'greater', 'need',
'for', 'more', 'accurate', 'aerodynamic', 'theories', 'either',
'to', 'predict', 'what', 'will', 'happen', 'in', 'unfamiliar',
'flight', 'conditions', 'or', 'to', 'effect', 'an',
'extrapolation', 'from', 'a', 'known', 'test', 'result', 'to',
'the', 'design', 'condition', '.']

after removal of stopwords:
['supersonic', 'flow', 'around', 'blunt', 'bodies', '.',
'newtonian', 'theory', 'impact', 'shown', 'useful', 'pressure',
'calculations', 'forward', 'facing', 'part', 'bodies',
'moving', 'high', 'speed', '.', 'familiar', 'practice', 'use',
'information', 'calculate', 'nonviscous', 'velocities', 'wall',
'estimate', 'rates', 'heat', 'transfer', '.', 'procedure',
'perhaps', 'open', 'question', ',', '.', 'heat-transfer',
'rates', 'depend', 'velocity', 'gradients', 'given',
'newtonian', 'analysis', '.', 'one', 'obtain', 'information',
'boundary-layer', 'stability', 'body', 'stability',
'derivatives', '.', 'seems', ',', 'therefore', ',',
'inevitable', ',', 'design', 'proceeds', 'hypersonic',
'missiles', ',', 'greater', 'need', 'accurate', 'aerodynamic',
'theories', 'either', 'predict', 'happen', 'unfamiliar',
'flight', 'conditions', 'effect', 'extrapolation', 'known',
'test', 'result', 'design', 'condition', '.']

before removing punctuations:

['supersonic', 'flow', 'around', 'blunt', 'bodies', '.',
'newtonian', 'theory', 'impact', 'shown', 'useful', 'pressure',

'calculations', 'forward', 'facing', 'part', 'bodies',
'moving', 'high', 'speed', '.', 'familiar', 'practice', 'use',
'information', 'calculate', 'nonviscous', 'velocities', 'wall',
'estimate', 'rates', 'heat', 'transfer', '.', 'procedure',
'perhaps', 'open', 'question', ',', '.', 'heat-transfer',
'rates', 'depend', 'velocity', 'gradients', 'given',
'newtonian', 'analysis', '.', 'one', 'obtain', 'information',
'boundary-layer', 'stability', 'body', 'stability',
'derivatives', '.', 'seems', ',', 'therefore', ',',
'inevitable', ',', 'design', 'proceeds', 'hypersonic',
'missiles', ',', 'greater', 'need', 'accurate', 'aerodynamic',
'theories', 'either', 'predict', 'happen', 'unfamiliar',
'flight', 'conditions', 'effect', 'extrapolation', 'known',
'test', 'result', 'design', 'condition', '.']

after removal of punctuations:

['supersonic', 'flow', 'around', 'blunt', 'bodies',
'newtonian', 'theory', 'impact', 'shown', 'useful', 'pressure',
'calculations', 'forward', 'facing', 'part', 'bodies',
'moving', 'high', 'speed', 'familiar', 'practice', 'use',
'information', 'calculate', 'nonviscous', 'velocities', 'wall',
'estimate', 'rates', 'heat', 'transfer', 'procedure',
'perhaps', 'open', 'question', 'heattransfer', 'rates',
'depend', 'velocity', 'gradients', 'given', 'newtonian',
'analysis', 'one', 'obtain', 'information', 'boundarylayer',
'stability', 'body', 'stability', 'derivatives', 'seems',
'therefore', 'inevitable', 'design', 'proceeds', 'hypersonic',
'missiles', 'greater', 'need', 'accurate', 'aerodynamic',
'theories', 'either', 'predict', 'happen', 'unfamiliar',
'flight', 'conditions', 'effect', 'extrapolation', 'known',
'test', 'result', 'design', 'condition']

2. **cranfield0041:**

before lower case:

['on', 'transition', 'experiments', 'at', 'moderate',
'supersonic', 'speeds', '.', 'studies', 'of', 'transition',
'over', 'a', 'flat', 'plate', 'at', 'mach', 'number', '1.76',
'were', 'carried', 'out', 'using', 'a', 'hot-wire',
'anemometer', 'as', 'one', 'of', 'the', 'principal', 'tools',
'.', 'the', 'nature', 'and', 'measurements', 'of',
'free-stream', 'disturbances', 'at', 'supersonic', 'speeds',
'are', 'analyzed', '.', 'the', 'experimental', 'results',

```
'are', 'interpreted', 'in', 'the', 'light', 'of', 'present',
'overall', 'information', 'on', 'transition', 'at',
'supersonic', 'speeds', 'and', 'conclusions', 'as', 'to',
'further', 'fruitful', 'experiments', 'are', 'drawn', '.']
```

after lowercase:

```
['on', 'transition', 'experiments', 'at', 'moderate',
'supersonic', 'speeds', '.', 'studies', 'of', 'transition',
'over', 'a', 'flat', 'plate', 'at', 'mach', 'number', '1.76',
'were', 'carried', 'out', 'using', 'a', 'hot-wire',
'anemometer', 'as', 'one', 'of', 'the', 'principal', 'tools',
'.', 'the', 'nature', 'and', 'measurements', 'of',
'free-stream', 'disturbances', 'at', 'supersonic', 'speeds',
'are', 'analyzed', '.', 'the', 'experimental', 'results',
'are', 'interpreted', 'in', 'the', 'light', 'of', 'present',
'overall', 'information', 'on', 'transition', 'at',
'supersonic', 'speeds', 'and', 'conclusions', 'as', 'to',
'further', 'fruitful', 'experiments', 'are', 'drawn', '.']
```

before removal of stopwords:

```
['on', 'transition', 'experiments', 'at', 'moderate',
'supersonic', 'speeds', '.', 'studies', 'of', 'transition',
'over', 'a', 'flat', 'plate', 'at', 'mach', 'number', '1.76',
'were', 'carried', 'out', 'using', 'a', 'hot-wire',
'anemometer', 'as', 'one', 'of', 'the', 'principal', 'tools',
'.', 'the', 'nature', 'and', 'measurements', 'of',
'free-stream', 'disturbances', 'at', 'supersonic', 'speeds',
'are', 'analyzed', '.', 'the', 'experimental', 'results',
'are', 'interpreted', 'in', 'the', 'light', 'of', 'present',
'overall', 'information', 'on', 'transition', 'at',
'supersonic', 'speeds', 'and', 'conclusions', 'as', 'to',
'further', 'fruitful', 'experiments', 'are', 'drawn', '.']
```

after removal of stopwords:

```
['transition', 'experiments', 'moderate', 'supersonic',
'speeds', '.', 'studies', 'transition', 'flat', 'plate',
'mach', 'number', '1.76', 'carried', 'using', 'hot-wire',
'anemometer', 'one', 'principal', 'tools', '.', 'nature',
'measurements', 'free-stream', 'disturbances', 'supersonic',
'speeds', 'analyzed', '.', 'experimental', 'results',
'interpreted', 'light', 'present', 'overall', 'information',
```

'transition', 'supersonic', 'speeds', 'conclusions',
'fruitful', 'experiments', 'drawn', '.']

before removing punctuations:

['transition', 'experiments', 'moderate', 'supersonic',
'speeds', '.', 'studies', 'transition', 'flat', 'plate',
'mach', 'number', '1.76', 'carried', 'using', 'hot-wire',
'anemometer', 'one', 'principal', 'tools', '.', 'nature',
'measurements', 'free-stream', 'disturbances', 'supersonic',
'speeds', 'analyzed', '.', 'experimental', 'results',
'interpreted', 'light', 'present', 'overall', 'information',
'transition', 'supersonic', 'speeds', 'conclusions',
'fruitful', 'experiments', 'drawn', '.']

after removal of punctuations:

['transition', 'experiments', 'moderate', 'supersonic',
'speeds', 'studies', 'transition', 'flat', 'plate', 'mach',
'number', '176', 'carried', 'using', 'hotwire', 'anemometer',
'one', 'principal', 'tools', 'nature', 'measurements',
'freestream', 'disturbances', 'supersonic', 'speeds',
'analyzed', 'experimental', 'results', 'interpreted', 'light',
'present', 'overall', 'information', 'transition',
'supersonic', 'speeds', 'conclusions', 'fruitful',
'experiments', 'drawn']

3. **cranfield0067:**

before lower case:

['dynamic', 'stability', 'of', 'vehicles', 'traversing',
'ascending', 'or', 'descending', 'paths', 'through', 'the',
'atmosphere', '.', 'an', 'analysis', 'is', 'given', 'of',
'the', 'oscillatory', 'motions', 'of', 'vehicles', 'which',
'traverse', 'ascending', 'and', 'descending', 'paths',
'through', 'the', 'atmosphere', 'at', 'high', 'speed', '.',
'the', 'specific', 'case', 'of', 'a', 'skip', 'path', 'is',
'examined', 'in', 'detail', ',', 'and', 'this', 'leads', 'to',
'a', 'form', 'of', 'solution', 'for', 'the', 'oscillatory',
'motion', 'which', 'should', 'recur', 'over', 'any',
'trajectory', '.', 'the', 'distinguishing', 'feature', 'of',
'this', 'form', 'is', 'the', 'appearance', 'of', 'the',
'bessel', 'rather', 'than', 'the', 'trigonometric', 'function',

```
'as', 'the', 'characteristic', 'mode', 'of', 'oscillation',
'.']
```

after lowercase:

```
['dynamic', 'stability', 'of', 'vehicles', 'traversing',
'ascending', 'or', 'descending', 'paths', 'through', 'the',
'atmosphere', '.', 'an', 'analysis', 'is', 'given', 'of',
'the', 'oscillatory', 'motions', 'of', 'vehicles', 'which',
'traverse', 'ascending', 'and', 'descending', 'paths',
'through', 'the', 'atmosphere', 'at', 'high', 'speed', '.',
'the', 'specific', 'case', 'of', 'a', 'skip', 'path', 'is',
'examined', 'in', 'detail', ',', 'and', 'this', 'leads', 'to',
'a', 'form', 'of', 'solution', 'for', 'the', 'oscillatory',
'motion', 'which', 'should', 'recur', 'over', 'any',
'trajectory', '.', 'the', 'distinguishing', 'feature', 'of',
'this', 'form', 'is', 'the', 'appearance', 'of', 'the',
'bessel', 'rather', 'than', 'the', 'trigonometric', 'function',
'as', 'the', 'characteristic', 'mode', 'of', 'oscillation',
'.']
```

before removal of stopwords:

```
['dynamic', 'stability', 'of', 'vehicles', 'traversing',
'ascending', 'or', 'descending', 'paths', 'through', 'the',
'atmosphere', '.', 'an', 'analysis', 'is', 'given', 'of',
'the', 'oscillatory', 'motions', 'of', 'vehicles', 'which',
'traverse', 'ascending', 'and', 'descending', 'paths',
'through', 'the', 'atmosphere', 'at', 'high', 'speed', '.',
'the', 'specific', 'case', 'of', 'a', 'skip', 'path', 'is',
'examined', 'in', 'detail', ',', 'and', 'this', 'leads', 'to',
'a', 'form', 'of', 'solution', 'for', 'the', 'oscillatory',
'motion', 'which', 'should', 'recur', 'over', 'any',
'trajectory', '.', 'the', 'distinguishing', 'feature', 'of',
'this', 'form', 'is', 'the', 'appearance', 'of', 'the',
'bessel', 'rather', 'than', 'the', 'trigonometric', 'function',
'as', 'the', 'characteristic', 'mode', 'of', 'oscillation',
'.']
```

after removal of stopwords:

```
['dynamic', 'stability', 'vehicles', 'traversing', 'ascending',
'descending', 'paths', 'atmosphere', '.', 'analysis', 'given',
'oscillatory', 'motions', 'vehicles', 'traverse', 'ascending',
'descending', 'paths', 'atmosphere', 'high', 'speed', '.',
```

'specific', 'case', 'skip', 'path', 'examined', 'detail', ',',
'leads', 'form', 'solution', 'oscillatory', 'motion', 'recur',
'trajectory', '.', 'distinguishing', 'feature', 'form',
'appearance', 'bessel', 'rather', 'trigonometric', 'function',
'characteristic', 'mode', 'oscillation', '.']

before removing punctuations:

['dynamic', 'stability', 'vehicles', 'traversing', 'ascending',
'descending', 'paths', 'atmosphere', '.', 'analysis', 'given',
'oscillatory', 'motions', 'vehicles', 'traverse', 'ascending',
'descending', 'paths', 'atmosphere', 'high', 'speed', '.',
'specific', 'case', 'skip', 'path', 'examined', 'detail', ',',
'leads', 'form', 'solution', 'oscillatory', 'motion', 'recur',
'trajectory', '.', 'distinguishing', 'feature', 'form',
'appearance', 'bessel', 'rather', 'trigonometric', 'function',
'characteristic', 'mode', 'oscillation', '.']

after removal of punctuations:

['dynamic', 'stability', 'vehicles', 'traversing', 'ascending',
'descending', 'paths', 'atmosphere', 'analysis', 'given',
'oscillatory', 'motions', 'vehicles', 'traverse', 'ascending',
'descending', 'paths', 'atmosphere', 'high', 'speed',
'specific', 'case', 'skip', 'path', 'examined', 'detail',
'leads', 'form', 'solution', 'oscillatory', 'motion', 'recur',
'trajectory', 'distinguishing', 'feature', 'form',
'appearance', 'bessel', 'rather', 'trigonometric', 'function',
'characteristic', 'mode', 'oscillation']

4. **cranfield0071:**
before lower case:

['laminar', 'boundary', 'layer', 'behind', 'shock',
'advancing', 'into', 'stationary', 'fluid', '.', 'a', 'study',
'was', 'made', 'of', 'the', 'laminar', 'compressible',
'boundary', 'layer', 'induced', 'by', 'a', 'shock', 'wave',
'advancing', 'into', 'a', 'stationary', 'fluid', 'bounded',
'by', 'a', 'wall', '.', 'for', 'weak', 'shock', 'waves', ',',
'the', 'boundary', 'layer', 'is', 'identical', 'with', 'that',
'which', 'occurs', 'when', 'an', 'infinite', 'wall', 'is',
'impulsively', 'set', 'into', 'uniform', 'motion', 'shocks',
'.', 'velocity', 'and', 'temperature', 'profiles', ',',
'recovery', 'factors', ',', 'and', 'skinfriction', 'and',

```
'heat-transfer', 'coefficients', 'are', 'tabulated', 'for',
'a', 'wide', 'range', 'of', 'shock', 'strengths', '.']
```

after lowercase:

```
['laminar', 'boundary', 'layer', 'behind', 'shock',
'advancing', 'into', 'stationary', 'fluid', '.', 'a', 'study',
'was', 'made', 'of', 'the', 'laminar', 'compressible',
'boundary', 'layer', 'induced', 'by', 'a', 'shock', 'wave',
'advancing', 'into', 'a', 'stationary', 'fluid', 'bounded',
'by', 'a', 'wall', '.', 'for', 'weak', 'shock', 'waves', ',',
'the', 'boundary', 'layer', 'is', 'identical', 'with', 'that',
'which', 'occurs', 'when', 'an', 'infinite', 'wall', 'is',
'impulsively', 'set', 'into', 'uniform', 'motion', 'shocks',
'.', 'velocity', 'and', 'temperature', 'profiles', ',',
'recovery', 'factors', ',', 'and', 'skinfriction', 'and',
'heat-transfer', 'coefficients', 'are', 'tabulated', 'for',
'a', 'wide', 'range', 'of', 'shock', 'strengths', '.']
```

before removal of stopwords:

```
['laminar', 'boundary', 'layer', 'behind', 'shock',
'advancing', 'into', 'stationary', 'fluid', '.', 'a', 'study',
'was', 'made', 'of', 'the', 'laminar', 'compressible',
'boundary', 'layer', 'induced', 'by', 'a', 'shock', 'wave',
'advancing', 'into', 'a', 'stationary', 'fluid', 'bounded',
'by', 'a', 'wall', '.', 'for', 'weak', 'shock', 'waves', ',',
'the', 'boundary', 'layer', 'is', 'identical', 'with', 'that',
'which', 'occurs', 'when', 'an', 'infinite', 'wall', 'is',
'impulsively', 'set', 'into', 'uniform', 'motion', 'shocks',
'.', 'velocity', 'and', 'temperature', 'profiles', ',',
'recovery', 'factors', ',', 'and', 'skinfriction', 'and',
'heat-transfer', 'coefficients', 'are', 'tabulated', 'for',
'a', 'wide', 'range', 'of', 'shock', 'strengths', '.']
```

after removal of stopwords:

```
['laminar', 'boundary', 'layer', 'behind', 'shock',
'advancing', 'stationary', 'fluid', '.', 'study', 'made',
'laminar', 'compressible', 'boundary', 'layer', 'induced',
'shock', 'wave', 'advancing', 'stationary', 'fluid', 'bounded',
'wall', '.', 'weak', 'shock', 'waves', ',', 'boundary',
'layer', 'identical', 'occurs', 'infinite', 'wall',
'impulsively', 'set', 'uniform', 'motion', 'shocks', '.',
'velocity', 'temperature', 'profiles', ',', 'recovery',
```

'factors', ',', 'skinfriction', 'heat-transfer',
'coefficients', 'tabulated', 'wide', 'range', 'shock',
'strengths', '.']

before removing punctuations:

['laminar', 'boundary', 'layer', 'behind', 'shock',
'advancing', 'stationary', 'fluid', '.', 'study', 'made',
'laminar', 'compressible', 'boundary', 'layer', 'induced',
'shock', 'wave', 'advancing', 'stationary', 'fluid', 'bounded',
'wall', '.', 'weak', 'shock', 'waves', ',', 'boundary',
'layer', 'identical', 'occurs', 'infinite', 'wall',
'impulsively', 'set', 'uniform', 'motion', 'shocks', '.',
'velocity', 'temperature', 'profiles', ',', 'recovery',
'factors', ',', 'skinfriction', 'heat-transfer',
'coefficients', 'tabulated', 'wide', 'range', 'shock',
'strengths', '.']

after removal of punctuations:

['laminar', 'boundary', 'layer', 'behind', 'shock',
'advancing', 'stationary', 'fluid', 'study', 'made', 'laminar',
'compressible', 'boundary', 'layer', 'induced', 'shock',
'wave', 'advancing', 'stationary', 'fluid', 'bounded', 'wall',
'weak', 'shock', 'waves', 'boundary', 'layer', 'identical',
'occurs', 'infinite', 'wall', 'impulsively', 'set', 'uniform',
'motion', 'shocks', 'velocity', 'temperature', 'profiles',
'recovery', 'factors', 'skinfriction', 'heattransfer',
'coefficients', 'tabulated', 'wide', 'range', 'shock',
'strengths']

5. **cranfield0098:**

before lower case:

['heat', 'transfer', 'by', 'laminar', 'flow', 'to', 'a',
'rotating', 'plate', '.', 'an', 'exact', 'solution', 'of',
'the', 'heat-transfer', 'problem', 'for', 'the', 'von',
'karman', 'example', 'of', 'the', 'laminar', 'flow', 'of', 'a',
'viscous', 'fluid', 'over', 'a', 'rotating', 'plate', 'is',
'given', 'in', 'dimensionless', 'form', 'and', 'physically',
'discussed', '.', 'the', 'solution', 'is', 'explicitly',
'given', 'for', 'a', 'constant', 'temperature', 'on', 'the',
'plate', 'with', 'viscous', 'dissipation', 'included', '.',
'the', 'numerical', 'results', 'are', 'given', 'for',
'prandtl', 'numbers', 'from', '0.5', 'to', '10', '.']

after lowercase:

```
['heat', 'transfer', 'by', 'laminar', 'flow', 'to', 'a',
'rotating', 'plate', '.', 'an', 'exact', 'solution', 'of',
'the', 'heat-transfer', 'problem', 'for', 'the', 'von',
'karman', 'example', 'of', 'the', 'laminar', 'flow', 'of', 'a',
'viscous', 'fluid', 'over', 'a', 'rotating', 'plate', 'is',
'given', 'in', 'dimensionless', 'form', 'and', 'physically',
'discussed', '.', 'the', 'solution', 'is', 'explicitly',
'given', 'for', 'a', 'constant', 'temperature', 'on', 'the',
'plate', 'with', 'viscous', 'dissipation', 'included', '.',
'the', 'numerical', 'results', 'are', 'given', 'for',
'prandtl', 'numbers', 'from', '0.5', 'to', '10', '.']
```

before removal of stopwords:

```
['heat', 'transfer', 'by', 'laminar', 'flow', 'to', 'a',
'rotating', 'plate', '.', 'an', 'exact', 'solution', 'of',
'the', 'heat-transfer', 'problem', 'for', 'the', 'von',
'karman', 'example', 'of', 'the', 'laminar', 'flow', 'of', 'a',
'viscous', 'fluid', 'over', 'a', 'rotating', 'plate', 'is',
'given', 'in', 'dimensionless', 'form', 'and', 'physically',
'discussed', '.', 'the', 'solution', 'is', 'explicitly',
'given', 'for', 'a', 'constant', 'temperature', 'on', 'the',
'plate', 'with', 'viscous', 'dissipation', 'included', '.',
'the', 'numerical', 'results', 'are', 'given', 'for',
'prandtl', 'numbers', 'from', '0.5', 'to', '10', '.']
```

after removal of stopwords:

```
['heat', 'transfer', 'laminar', 'flow', 'rotating', 'plate',
'.', 'exact', 'solution', 'heat-transfer', 'problem', 'von',
'karman', 'example', 'laminar', 'flow', 'viscous', 'fluid',
'rotating', 'plate', 'given', 'dimensionless', 'form',
'physically', 'discussed', '.', 'solution', 'explicitly',
'given', 'constant', 'temperature', 'plate', 'viscous',
'dissipation', 'included', '.', 'numerical', 'results',
'given', 'prandtl', 'numbers', '0.5', '10', '.']
```

before removing punctuations:

```
['heat', 'transfer', 'laminar', 'flow', 'rotating', 'plate',
'.', 'exact', 'solution', 'heat-transfer', 'problem', 'von',
'karman', 'example', 'laminar', 'flow', 'viscous', 'fluid',
```

```
'rotating', 'plate', 'given', 'dimensionless', 'form',
'physically', 'discussed', '.', 'solution', 'explicitly',
'given', 'constant', 'temperature', 'plate', 'viscous',
'dissipation', 'included', '.', 'numerical', 'results',
'given', 'prandtl', 'numbers', '0.5', '10', '.']
```

after removal of punctuations:

```
['heat', 'transfer', 'laminar', 'flow', 'rotating', 'plate',
'exact', 'solution', 'heattransfer', 'problem', 'von',
'karman', 'example', 'laminar', 'flow', 'viscous', 'fluid',
'rotating', 'plate', 'given', 'dimensionless', 'form',
'physically', 'discussed', 'solution', 'explicitly', 'given',
'constant', 'temperature', 'plate', 'viscous', 'dissipation',
'included', 'numerical', 'results', 'given', 'prandtl',
'numbers', '05', '10']
```

TF-IDF Matrix:

Steps followed:

A. For documents:

1. Collected all the words in a dictionary set containing the corpus of words present in all the components
2. Made a `**word**` object that had 5 dictionaries corresponding to it. Where the key was **document-ID** and the value was the corresponding **TF of that word**. Had a **df** variable as well which contained the document frequency for that word
3. Created a matrix of **1400** x **8971** where 1400 is the number of total documents and 8971 was the vocabulary size.
4. Made 5 such matrices, one for each type of tf-weighting scheme
   Sample word class:

```
word: aerodynamics
```

```
binary tf dictionary : {244: 1, 216: 1, 289: 1, 1: 1, 360: 1,
225: 1, 11: 1, 284: 1, 297: 1, 689: 1, 453: 1, 634: 1, 1380: 1,
1347: 1, 1206: 1, 1271: 1, 1331: 1, 237: 1, 296: 1, 753: 1, 792:
1, 902: 1, 685: 1}
```

```
Document freq: 23
```

```
log norm tf dictionary: {244: 0.6931471805599453, 216:
0.6931471805599453, 289: 0.6931471805599453, 1:
```

0.6931471805599453, 360: 0.6931471805599453, 225:
0.6931471805599453, 11: 0.6931471805599453, 284:
0.6931471805599453, 297: 0.6931471805599453, 689:
0.6931471805599453, 453: 0.6931471805599453, 634:
1.0986122886681098, 1380: 0.6931471805599453, 1347:
0.6931471805599453, 1206: 0.6931471805599453, 1271:
0.6931471805599453, 1331: 0.6931471805599453, 237:
0.6931471805599453, 296: 0.6931471805599453, 753:
1.0986122886681098, 792: 0.6931471805599453, 902:
0.6931471805599453, 685: 0.6931471805599453}

count tf dictionary: {244: 1, 216: 1, 289: 1, 1: 1, 360: 1, 225:
1, 11: 1, 284: 1, 297: 1, 689: 1, 453: 1, 634: 2, 1380: 1, 1347:
1, 1206: 1, 1271: 1, 1331: 1, 237: 1, 296: 1, 753: 2, 792: 1,
902: 1, 685: 1}

freq tf dictionary: {244: 0.0038461538461538464, 216:
0.006756756756756757, 289: 0.007518796992481203, 1:
0.01282051282051282, 360: 0.012048192771084338, 225:
0.004608294930875576, 11: 0.016129032258064516, 284:
0.017857142857142856, 297: 0.012048192771084338, 689:
0.006711409395973154, 453: 0.008264462809917356, 634:
0.024390243902439025, 1380: 0.006211180124223602, 1347:
0.006711409395973154, 1206: 0.013513513513513514, 1271:
0.005988023952095809, 1331: 0.018518518518518517, 237:
0.012987012987012988, 296: 0.007518796992481203, 753:
0.015037593984962405, 792: 0.00398406374501992, 902:
0.008620689655172414, 685: 0.005813953488372093}

double norm tf dictionary: {244: 0.53125, 216:
0.5555555555555556, 289: 0.6, 1: 0.6, 360: 0.6, 225:
0.5384615384615384, 11: 0.6666666666666666, 284:
0.6666666666666666, 297: 0.6666666666666666, 689: 0.55, 453:
0.5833333333333334, 634: 0.75, 1380: 0.5555555555555556, 1347:
0.5555555555555556, 1206: 0.6666666666666666, 1271:
0.5833333333333334, 1331: 0.625, 237: 0.625, 296:
0.5833333333333334, 753: 0.6, 792: 0.5833333333333334, 902:
0.625, 685: 0.5833333333333334}

B. For query

1. Preprocessed the query using the same methods that were applied on the documents
2. Created the TF vector for the query. 5 different types of vectors were made, one for each type of TF weighting scheme
3. Took the dot product of this vector with every tf-idf vector of each document and stored the result in a dictionary as (doc-id : dot product).
4. Sorted this resultant dictionary on the basis of value. When values were the same, the insertion order was considered.

Note: We have not normalized the TF vectors for documents since the norm was very small for some vectors and hence became NaN.

TF weighting schemes:

1. Binary weighting:
    a. Pros:
        i. It is Computationally beneficial since we are only storing binary values hence instead of a float dtype we can use Bool datatype which saves memory and also helps in faster computation
        ii. Can help cancel out any noise that is present in the data since we are only checking for a membership instance and hence does not give more importance to a word that occurs more than it should.
    b. Cons:
        i. Cannot comprehend the complete document and hence leads to a loss of information as the count is not captured and hence the importance is not captured fully for a word.
        ii. Cannot fully capture the emphasis that a word could have in a document
2. Count Weighting:
    a. Pros:
        i. Can capture the frequency of each word and hence can impart information about the presence of that word in the document and can also help in quantifying it's importance
        ii. Is very beneficial for models where count is required such as count-sensitive information retrieval
    b. Cons:
        i. Since the document length is never considered the values are absolute and never normalised and hence some words can have inflated count values which can lead to wrong interpretation of importance
        ii. Some words that may be similar in meaning may have very different count values which may lead to one synonym getting more importance than the other which is not desirable
3. Frequency Weighting:
    a. Pros:

        i. It can correctly represent the importance of each word in a document since the tf values are now normalised to encapsulate the complete vocabulary of a document
        ii. Can be a very useful tool for represent similar tf -weights in an n-dimensional space using these weights as similar importance words may have similar tf weights
    b. Cons:
        i. Since it is normalised using the document length, some documents with larger noise or useless words can totally dominate the other documents for relevance. This attribute is undesirable because it can lead to false positives during retrieval
        ii. Since it is a simple division, it can give more weight to words that appear too much in a document which has a large vocabulary and this can lead to a linear increase in it's importance which is not always required since documents have been proven to contain words in a log(x) kind of pattern in terms of their importance where x is their frequency.

4. Log Normalised TF:
    a. Pros:
        i. Since it has a log function it can help in dampening the freq of each word and hence impart a more imbalance weight distribution wherein a word that occurs less frequently is given a weight that is comparable to a word that occurs exponentially more than the former.
        ii. Can be useful where there is a large corpus of words present as it can normalise to correct for the importance of words within the document
    b. Cons:
        i. Since log(x) has a very large slope within the vicinity of 1, it can often lead to very distorted weights and hence leads to a poorly weighted matrix in some cases
5. Double normalised TF:
    a. Pros:
        i. Can help impart proportionate importance to words in the manner they occur in the document and hence is the most suitable for many applications
        ii. Since it does double normalization the tf weights can become resistant to noise and hence is a more accurate representation of the vocabulary words
    b. Cons:
        i. Since it includes floating point arithmetic it becomes computationally expensive for very large documents and memory inefficient to store as well
        ii. Double normalization often results in over smoothing of the input data and hence can lead to possible loss of information

Jaccard Index:

For documents:

1. Made a set of the words present in a document and the set only contained unique instances of each word.
2. Made a dictionary for doc_id to it's corresponding vocabulary.

For query:

1. Preprocessed the query in the similar way as the documents
2. Calculated numerator as the number of elements in the set containing the words that were present in the query and the document
3. Calculated the denominator as the number of words in the set of union of documents and query.
4. Calculated the Intersection over Union for each document and stored in a dictionary. Then sorted the dictionary on the basis of this value.
5. Printed top-10 results based on these values for each query.
   Example query:

```
Enter number of queries: 3

Enter query: velocity temperature aerodynamics
[(407, 0.06896551724137931), (378, 0.05263157894736842), (549,
0.05263157894736842), (71, 0.05128205128205128), (31, 0.05), (269,
0.04878048780487805), (61, 0.047619047619047616), (154,
0.047619047619047616), (460, 0.0425531914893617), (485,
0.041666666666666664)]

Enter query: complete

[(684, 0.03225806451612903), (1038, 0.025), (1400,
0.022222222222222223), (372, 0.021739130434782608), (1078,
0.019230769230769232), (1135, 0.019230769230769232), (702,
0.018867924528301886), (319, 0.018518518518518517), (55,
0.017543859649122806), (1091, 0.017241379310344827)]

Enter query: entry

[(1348, 0.02702702702702703), (715, 0.022727272727272728), (1345,
0.018867924528301886), (944, 0.018518518518518517), (967,
0.01818181818181818), (275, 0.017543859649122806), (1394,
0.016666666666666666), (1346, 0.01639344262295082), (69,
0.015873015873015872), (982, 0.015873015873015872)]
```

Q2)

1. The dataset is preprocessed. The dataset is in CSV format with the columns ‘ArticleID’, ‘text’ and ‘category’. The text is cleaned by removing punctuations, stop words, and turning it into lower case. The text is then tokenized, and the tokens are then stemmed.
   TCF-IF weighing scheme is then implemented on these stemmed tokens.
2. Used sklearn in build test_train_split function to split the dataset into training and testing.
3. Training the Naive Bayes Model.

   In order to create a Naive Bayes Model, we calculated the probability of each category.
   We calculated this simply using the formula:
   no of documents having the given category/ total documents

   We also need to calculate the probability of each feature given each category based on the TF-ICF values of that feature in documents belonging to that category. We used the TF-ICF Matrix built above. We **assumed** that this probability will be equal to:
   TF-ICF score of the term given a category/ Sum of all TF-ICF scores of the terms for the given category, present in the document belonging to that category.

   Now, we need to predict the category for any given query. To find this, we calculate the probability of the category being C given this query, where C is any category. We find this for all the categories, and the category having maximum probability is taken as the predicted output. We used Naive Bayes concepts to find this probability.

   Assumptions:
   While calculating the probability, instead of taking individual terms probability of belonging to a particular category, we added 1 to it and took its logarithm. The same is done while using the probability of each category. This is done to normalize and handle the case of any probability being zero.

Thus at the end, we compared the log of the probability of the category being C given a query.

Log P(Category| given query)= log (Category+1) + Summation(log(P(Term|Category)+1)

4. We used Sklearn's inbuilt functions like accuracy_score, recall_score, precision_score, f1_score to evaluate the model. We got the following results:
   Accuracy Score: 0.8098434004474273
   Precision Score: 0.8686597102167877
   Recall Score: 0.7933298327421372
   F1 Score: 0.8053600372586922
5. Using accuracy as the parameter:
   Original TF-ICF implementation, with 70-30 split and stemming: 0.809
   TF-ICF implementation with 80-20 split and stemming: 0.802
   TF-ICF implementation with 70-30 split and lemmatization: 0.79
   TF-ICF implementation with 80-20 split and lemmatization: 0.78

   TF-IDF implementation with 70-30 split: 0.92
   TF-IDF implementation with 80-20 split: 0.93
   TF-IDF implementation with 80-20 split and uni- and bigrams: 0.92
   TF-IDF implementation with 80-20 split and n-grams (n in 1:5): 0.91

6. With the experimentation, it would appear that the highest accuracy is of the TF-IDF Vectorisation (at 0.93). TF-IDF vectorisation appears to have a much better performance than TF-ICF. The main reason it works better than TF-ICF is that it considers not only the frequency of a term in a document (TF) but also how important the term is across the entire corpus (IDF). TF-ICF, on the other hand, only considers how important a term is in a specific class, based on its frequency in that class (ICF). This approach can lead to overfitting, where the model becomes too specific to the training data and performs poorly on new, unseen data. In contrast, TF-IDF takes into

account the frequency of a term across the entire corpus, which provides a more balanced weighting scheme that is less prone to overfitting.

It would also appear that using n-grams decreases the accuracy of the model. Using n-grams can increase the number of features (i.e., words or word combinations) in the dataset, which can lead to a higher-dimensional feature space. This can cause the model to become overfit to the training data, meaning that it fits the training data too closely and does not generalize well to new, unseen data. Additionally, using n-grams can increase the sparsity of the dataset, meaning that many of the features have very few occurrences in the dataset. This can make it difficult for the model to identify useful patterns in the data, as it may focus on noisy or irrelevant features.

Q3) For this part, the Microsoft Learning to Rank dataset is used. Specifically, only query qid=4 is considered. I have preprocessed the file to remove all qid which are not 0, to reduce the file size so that I can upload it on Google Drive and Colab properly.

For the first task, the maximum DCG (discounted cumulative gain) has to be calculated. This can be calculated by rearranging the query-url pairs by sorting them on their relevance score in the reverse order.
As the relevance scores are 0, 1, 2, 3, the number of such files is
(num_of_rel_score_0)! * (num_of_rel_score_1)! * (num_of_rel_score_2)! * (num_of_rel_score_3)!
= 59! * 26! * 17! * 1!
The DCG can be calculated as

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

The maximum DCG is 20.989750804831445

Then, the nDCG (normalised DCG) is calculated. The NDCG can be calculated as:

NDCG = DCG/IDCG,
Where IDCG is the Ideal DCG

For documents = 50, nDCG = 0.3521042740324887
For the entire dataset, nDCG = 0.5979226516897831

For the third task, we have calculated the precision and the recall score.
First, the matrix is sorted on the basis of the sum of TF-IDF on the whole document, which is stored in feature 75.
Then, the precision and recall are calculated by iterating over the entire dataset.

Precision = P(relevant|retrieved)
Recall = P(retrieved|relevant)

We get the following Precision-Recall Curve: