Project Report

News Summarization and classification

Files attached: -

- **News-summarization-using_tf-idf. ipynb**
- **svmclassification. Ipynb**
- **news_gui.py**
- **data.sav (containing dataset for classification)**
- **svm.sav (svm model used in GUI)**

By:

Name : **P VISHNU TEJ**

Roll-no. : **1910110263**

Email : **pt770@snu.edu.in**

Project overview

- Created a Python graphic user interface that could summarize and classify news by either providing URL link or direct text.
- For summarization used tf-idf scores to give sentences weightage and then summarize the entire text by printing those sentences which have more weightage in the document.
- For classification I trained a svm model on a BBC news dataset and the given graphic user interface can effectively classify the given text into technology, sports, politics, business and entertainment.
- Created a graphic user interface using the tkinter library of python

News Summarization

Term Frequency * Inverse Document Frequency

In a simple language, TF-IDF can be defined as follows:

A High weight in TF-IDF is reached by a high term frequency (in the given document) and a low document frequency of the term in the whole collection of documents.

TF-IDF algorithm is made of 2 algorithms multiplied together.

Term Frequency

Term frequency (TF) is how often a word appears in a document, divided by how many words there are.

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

Inverse document frequency

Inverse document frequency (IDF) is how unique or rare a word is.
IDF(t) = log_e(Total number of documents / No. of docs with term t in it)

Data pre-processing

Raw text has unwanted characters (\n,\t,$ etc) and contains stop words (a, an, the) which has to be removed before generating the vector representation. The following text pre-processing techniques have been used:

1. Converting to lower case
2. Removal of stop words
3. Tokenize
4. Lemmatization

We can implement summarization using 9 steps:

1. Tokenize the sentences

We tokenize the sentences here instead of words.


```
["Ink helps drive democracy in Asia\n\nThe Kyrgyz Republic, a small, mountainous state of the former Soviet republic, is using invisible ink and ultraviolet readers in the country's elections as part of a drive to prevent multiple voting.", 'This new tec hnology is causing both worries and guarded optimism among different sectors of the population.', 'In an effort to live up to i ts reputation in the 1990s as "an island of democracy", the Kyrgyz President, Askar Akaev, pushed through the law requiring the use of ink during the upcoming Parliamentary and Presidential elections.', 'The US government agreed to fund all expenses assoc iated with this decision.', 'The Kyrgyz Republic is seen by many experts as backsliding from the high point it reached in the m id-1990s with a hastily pushed through referendum in 2003, reducing the legislative branch to one chamber with 75 deputies.',
```


2. Create the Frequency matrix of the words in each sentence.

We calculate the frequency of words in each sentence. The result would be:


```
{'Ink helps drive': {'ink': 2, 'help': 1, 'drive': 2, 'democracy': 1, 'asia': 1, 'kyrgyz': 1, 'republic': 2, ',': 3, 'small': 1, 'mountainous': 1, 'state': 1, 'former': 1, 'soviet': 1, 'using': 1, 'invisible': 1, 'ultraviolet': 1, 'reader': 1, 'countr y': 1, "'s": 1, 'election': 1, 'part': 1, 'prevent': 1, 'multiple': 1, 'voting': 1, '.': 1}, 'This new techno': {'new': 1, 'tec hnology': 1, 'causing': 1, 'worry': 1, 'guarded': 1, 'optimism': 1, 'among': 1, 'different': 1, 'sector': 1, 'population': 1, '.': 1}, 'In an effort to': {'effort': 1, 'live': 1, 'reputation': 1, '1990s': 1, '``': 1, 'island': 1, 'democracy': 1, "''": 1, ',': 3, 'kyrgyz': 1, 'president': 1, 'askar': 1, 'akaev': 1, 'pushed': 1, 'law': 1, 'requiring': 1, 'use': 1, 'ink': 1, 'upc oming': 1, 'parliamentary': 1, 'presidential': 1, 'election': 1, '.': 1}, 'The US governme': {'u': 1, 'government': 1, 'agree d': 1, 'fund': 1, 'expense': 1, 'associated': 1, 'decision': 1, '.': 1}, 'The Kyrgyz Repu': {'kyrgyz': 1, 'republic': 1, 'see n': 1, 'many': 1, 'expert': 1, 'backsliding': 1, 'high': 1, 'point': 1, 'reached': 1, 'mid-1990s': 1, 'hastily': 1, 'pushed': 1, 'referendum': 1, '2003': 1, ',': 1, 'reducing': 1, 'legislative': 1, 'branch': 1, 'one': 1, 'chamber': 1, '75': 1, 'deputy': 1, '.': 1}, 'The use of ink ': {'use': 1, 'ink': 1, 'reader': 1, 'panacea': 1, 'election': 1, 'ill': 1, '.': 1}, 'The actual te
```

3. Calculate Term Frequency and generate a matrix

We'll find the Term Frequency for each word in a paragraph.

{'Ink helps drive': {'ink': 0.08, 'help': 0.04, 'drive': 0.08, 'democracy': 0.04, 'asia': 0.04, 'kyrgyz': 0.04, 'republic': 0.0
8, ',': 0.12, 'small': 0.04, 'mountainous': 0.04, 'state': 0.04, 'former': 0.04, 'soviet': 0.04, 'using': 0.04, 'invisible': 0.
04, 'ultraviolet': 0.04, 'reader': 0.04, 'country': 0.04, "'s": 0.04, 'election': 0.04, 'part': 0.04, 'prevent': 0.04, 'multipl
e': 0.04, 'voting': 0.04, '.': 0.04}, 'This new techno': {'new': 0.09090909090909091, 'technology': 0.09090909090909091, 'causi
ng': 0.09090909090909091, 'worry': 0.09090909090909091, 'guarded': 0.09090909090909091, 'optimism': 0.09090909090909091, 'amon
g': 0.09090909090909091, 'different': 0.09090909090909091, 'sector': 0.09090909090909091, 'population': 0.09090909090909091,
'.': 0.09090909090909091}, 'In an effort to': {'effort': 0.043478260869565216, 'live': 0.043478260869565216, 'reputation': 0.04
3478260869565216, '1990s': 0.043478260869565216, '``': 0.043478260869565216, 'island': 0.043478260869565216, 'democracy': 0.043
478260869565216, "''": 0.043478260869565216, ',': 0.13043478260869565, 'kyrgyz': 0.043478260869565216, 'president': 0.043478260
869565216, 'askar': 0.043478260869565216, 'akaev': 0.043478260869565216, 'pushed': 0.043478260869565216, 'law': 0.0434782608695
65216, 'requiring': 0.043478260869565216, 'use': 0.043478260869565216, 'ink': 0.043478260869565216, 'upcoming': 0.0434782608695
65216, 'parliamentary': 0.043478260869565216, 'presidential': 0.043478260869565216, 'election': 0.043478260869565216, '.': 0.04

4. Creating a table for documents per words

We calculate, "**how many sentences contain a particular word**", Let's call it Documents per words matrix.

{'ink': 18, 'help': 1, 'drive': 2, 'democracy': 2, 'asia': 1, 'kyrgyz': 3, 'republic': 3, ',': 17, 'small': 1, 'mountainous':
1, 'state': 1, 'former': 2, 'soviet': 2, 'using': 1, 'invisible': 2, 'ultraviolet': 3, 'reader': 3, 'country': 3, "'s": 4, 'ele
ction': 10, 'part': 2, 'prevent': 1, 'multiple': 1, 'voting': 1, '.': 30, 'new': 1, 'technology': 2, 'causing': 1, 'worry': 1,
'guarded': 1, 'optimism': 1, 'among': 1, 'different': 1, 'sector': 1, 'population': 2, 'effort': 1, 'live': 1, 'reputation': 1,
'1990s': 1, '``': 2, 'island': 1, "''": 3, 'president': 1, 'askar': 1, 'akaev': 1, 'pushed': 2, 'law': 2, 'requiring': 1, 'us
e': 7, 'upcoming': 2, 'parliamentary': 3, 'presidential': 2, 'u': 1, 'government': 1, 'agreed': 1, 'fund': 1, 'expense': 1, 'as
sociated': 2, 'decision': 1, 'seen': 1, 'many': 2, 'expert': 1, 'backsliding': 1, 'high': 1, 'point': 1, 'reached': 1, 'mid-199
0s': 1, 'hastily': 1, 'referendum': 1, '2003': 1, 'reducing': 1, 'legislative': 1, 'branch': 1, 'one': 3, 'chamber': 1, '75':
1, 'deputy': 1, 'panacea': 1, 'ill': 1, 'actual': 1, 'behind': 1, 'complicated': 1, 'sprayed': 2, 'person': 1, 'left': 2, 'thum

5. Calculate IDF and generate a matrix

We'll find the IDF for each word in a paragraph.

{'Ink helps drive': {'ink': 0.2498774732165999, 'help': 1.505149978319906, 'drive': 1.2041199826559248, 'democracy': 1.20411998
26559248, 'asia': 1.505149978319906, 'kyrgyz': 1.0280287236002434, 'republic': 1.0280287236002434, ',': 0.27470105694163205, 's
mall': 1.505149978319906, 'mountainous': 1.505149978319906, 'state': 1.505149978319906, 'former': 1.2041199826559248, 'soviet':
1.2041199826559248, 'using': 1.505149978319906, 'invisible': 1.2041199826559248, 'ultraviolet': 1.0280287236002434, 'reader':
1.0280287236002434, 'country': 1.0280287236002434, "'s": 0.9030899869919435, 'election': 0.505149978319906, 'part': 1.204119982
6559248, 'prevent': 1.505149978319906, 'multiple': 1.505149978319906, 'voting': 1.505149978319906, '.': 0.028028723600243534},
'This new techno': {'new': 1.505149978319906, 'technology': 1.2041199826559248, 'causing': 1.505149978319906, 'worry': 1.505149
978319906, 'guarded': 1.505149978319906, 'optimism': 1.505149978319906, 'among': 1.505149978319906, 'different': 1.505149978319
906, 'sector': 1.505149978319906, 'population': 1.2041199826559248, '.': 0.028028723600243534}, 'In an effort to': {'effort':

6. Calculate TF-IDF and generate a matrix

We multiply the values from both the term frequency matrix, idf matrix generating a new matrix.

{'Ink helps drive': {'ink': 0.01999019785732799, 'help': 0.06020599913279624, 'drive': 0.09632959861247399, 'democracy': 0.0481
64799306236995, 'asia': 0.06020599913279624, 'kyrgyz': 0.04112114894400974, 'republic': 0.08224229788801948, ',': 0.03296412683
299584, 'small': 0.06020599913279624, 'mountainous': 0.06020599913279624, 'state': 0.06020599913279624, 'former': 0.04816479930
6236995, 'soviet': 0.048164799306236995, 'using': 0.06020599913279624, 'invisible': 0.048164799306236995, 'ultraviolet': 0.0411
2114894400974, 'reader': 0.04112114894400974, 'country': 0.04112114894400974, "'s": 0.03612359947967774, 'election': 0.02020599
913279624, 'part': 0.048164799306236995, 'prevent': 0.06020599913279624, 'multiple': 0.06020599913279624, 'voting': 0.060205999
13279624, '.': 0.0011211489440097415}, 'This new techno': {'new': 0.13683181621090054, 'technology': 0.10946545296872044, 'caus
ing': 0.13683181621090054, 'worry': 0.13683181621090054, 'guarded': 0.13683181621090054, 'optimism': 0.13683181621090054, 'amon
g': 0.13683181621090054, 'different': 0.13683181621090054, 'sector': 0.13683181621090054, 'population': 0.10946545296872044,
'.': 0.002548065781840321}, 'In an effort to': {'effort': 0.0654413034052133, 'live': 0.0654413034052133, 'reputation': 0.06544

7. Score the sentences

Here, we are using Tf-IDF score of words in a sentence to give weight to the paragraph by adding the Tf-idf frequency of every non-stop word in a sentence divided by total no of words in a sentence.

{'Ink helps drive': 0.04944558212998766, 'This new techno': 0.11964850012786234, 'In an effort to': 0.049380656249305724, 'The
US governme': 0.16036013400274382, 'The Kyrgyz Repu': 0.056479122060062274, 'The use of ink ': 0.11186605701392763, 'The actual
tech': 0.1665965587342357, 'The ink is spra': 0.12494724185797523, 'It dries and is': 0.21081909545984895, 'However, the pr':
0.07134710006349634, 'At the entrance': 0.04864899801774233, 'If the ink show': 0.10160743531280843, 'Likewise, any v': 0.13367
935000121356, 'These elections': 0.04705437075603565, 'Widely circulat': 0.0670031723331395, 'The author of o': 0.1097490957900
5312, 'The greatest pa': 0.12607279078676817, 'Local newspaper': 0.06996110139599462, 'Others, such as': 0.09252949175449168,
'This type of in': 0.06908862442484746, 'The other commo': 0.07040660684200312, 'The use of "inv': 0.10792821134304274, 'In mos
t electio': 0.14787026927281943, 'In Serbia, for ': 0.0928220270275755, 'Other rumours a': 0.07659642556393663, 'However, in re
a': 0.09992030485021251, 'The ink stays o': 0.14074650111050466, 'The passage of ': 0.08205575549262804, "The country's w": 0.1
071704603204813, 'David Mikosz wo': 0.09960384972389645}

8. Finding the threshold

We're calculating the average sentence score. And assign it as threshold. Here I have assigned threshold value as 1.1*average score.

9. Generate the summary

We select a sentence for a summarization if the sentence score is more than the average score.

```
This new technology is causing both worries and guarded optimism among different sectors of the population. The US government
agreed to fund all expenses associated with this decision. The use of ink is only one part of a general effort to show commitme
nt towards more open elections - the German Embassy, the Soros Foundation and the Kyrgyz government have all contributed to pur
chase transparent ballot boxes. The actual technology behind the ink is not that complicated. The ink is sprayed on a person's
left thumb. It dries and is not visible under normal light. Likewise, any voter who refuses to be inked will not receive the ba
llot. The use of ink has been controversial - especially among groups perceived to be pro-government. The greatest part of the
opposition to ink has often been sheer ignorance. In most elections, numerous rumors have spread about it. The ink stays on the
finger for at least 72 hours and for up to a week. The use of ink and readers by itself is not a panacea for election ills.
```

News Classification

Steps:

- Load the data from the BBC news dataset
- We pre-process the data (given text)
- We prepare the train and test data sets
- The training data set will be used to fit the model and the predictions will be performed on the test data set. We used train_test_split from the sklearn library. The Training Data will have 70% of the corpus and Test data will have the remaining 30% as we have set the parameter test_size=0.3 .
- First, we fit the TF-IDF model on the whole corpus. This will help the TF-IDF build a vocabulary of words which it has learned from the data and it will assign a unique integer number to each of these words.
- There are a maximum of 5000 unique words/features as we have set parameter max_features=5000.

- Result of tf-idf vectorization using sklearn feature

```
(0, 4979)     0.037507480915259406
(0, 4962)     0.021779883973408805
(0, 4952)     0.05774831226007798
(0, 4949)     0.11390200445671901
(0, 4934)     0.0840061946016908
(0, 4887)     0.03105393653657155
(0, 4844)     0.03287780356363458
(0, 4831)     0.04888449794306824
(0, 4820)     0.07658866257943193
(0, 4745)     0.03737106518438833
(0, 4714)     0.08432688343999196
```

i) 0 part: Row number of 'Train_X_Tfidf

ii) Unique Integer number of each word in the first row,

iii) Score calculated by TF-IDF Vectorizer

- Now we feed our data sets to the svm classification Algorithms.
- Classification report of our svm model

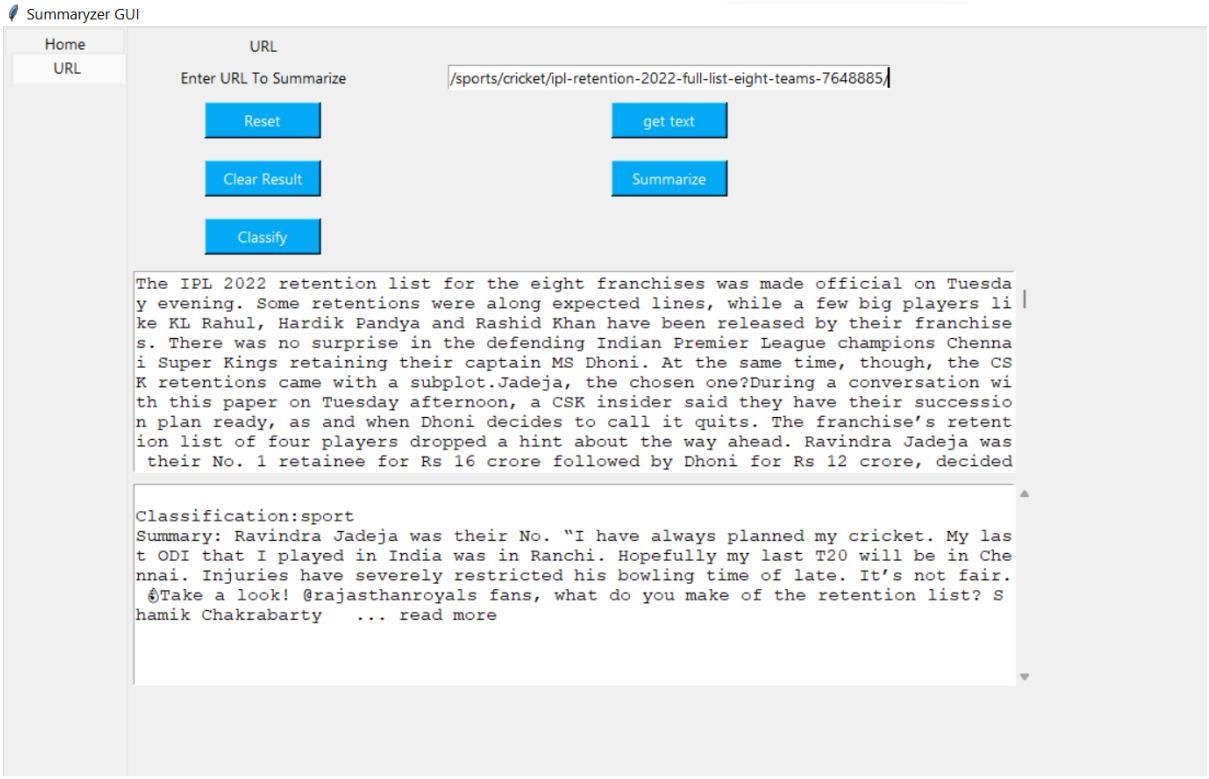
```
Accuracy:  0.9805389221556886
              precision    recall  f1-score   support

           0       0.97      0.97      0.97       158
           1       0.99      0.98      0.99       124
           2       0.96      0.96      0.96       117
           3       1.00      1.00      1.00       142
           4       0.98      0.99      0.98       127

    accuracy                           0.98       668
   macro avg       0.98      0.98      0.98       668
weighted avg       0.98      0.98      0.98       668
```

Sample outputs:

- URL ： https://indianexpress.com/article/sports/cricket/ipl-retention-2022-full-list-eight-teams-7648885/



- **URL: https://timesofindia.indiatimes.com/blogs/voices/key-technology-trends-that-will-shape-2022-and-beyond/**