DTI5125: Data Science Applications

Text Classification Group Assignment 1

Group Members:

Basma Reda Shaban Abd-Elsalam Abd-Elwahab

Amir Safwat Halim Youssef

Nada Ashraf Ismail AboElfetoh

Yasmine Ahmed Elsayed Mohamed

1. Overview

The main objective of this assignment is to build ML models (SVM, KNN, and decision trees) to predict the author of a book. I am also using many different packages and tools in Python and learning BOW, Tf-Idf, and n-grams transformations to represent the text as a vector. Learning how to choose a champion model and why use error analysis

2. Methodology

We followed some defined steps to obtain the aimed results:

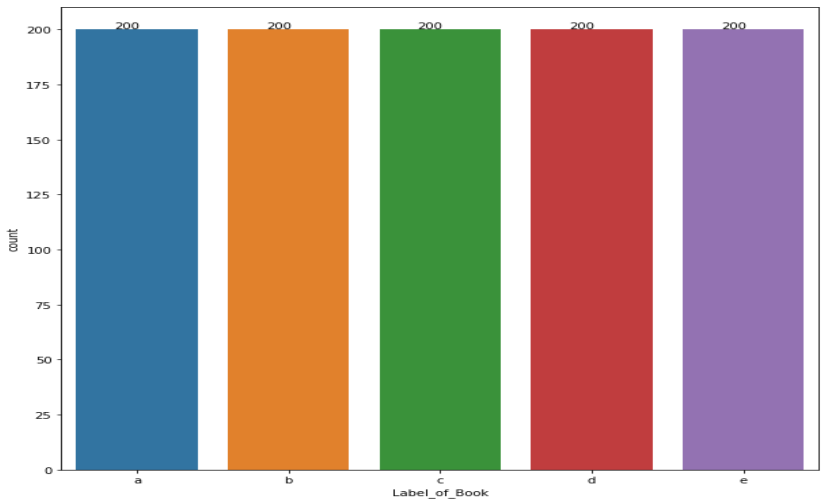
1. Cleaning data

```
from urllib import request
#for loop to get every book in BooksURLs list
for URL  in BooksURLs :
  response = request.urlopen(URL)
  raw = response.read().decode('utf8' , errors = 'replace')
  wordsList= re.findall(r"[a-zA-Z]{3,}", raw)
  #perform lemmetization on the data
  lemmatizer = WordNetLemmatizer()
  lemmitizedWords =[]
  for i in wordsList:
    words = i.lower()
    word = lemmatizer.lemmatize(words)
    #check if the word not in stopwords set
    if word not in set(stopwords.words('english')):
      lemmitizedWords.append(str(word))
  Books.append(lemmitizedWords)
```

2. Partitioning Data

- Every book have 200 partition, and every partition have 100 words.
- Using CountPlot to ensure that every book have 200 partitions

```
import seaborn as sns
plt.figure(figsize=(10,10))
ax = sns.countplot(x=result["Label_of_Book"],  data=result, order = result["Label_of_Book"].value_counts().index )
for p, label in zip(ax.patches, result["Label_of_Book"].value_counts()):
    ax.annotate(label, (p.get_x()+0.25, p.get_height()+0.5))
```

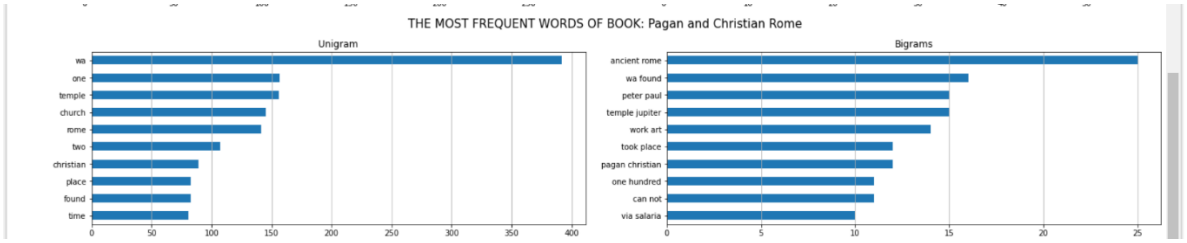


- **Count the most frequented words using two methods: Unigram, Bigram and WordCloud**

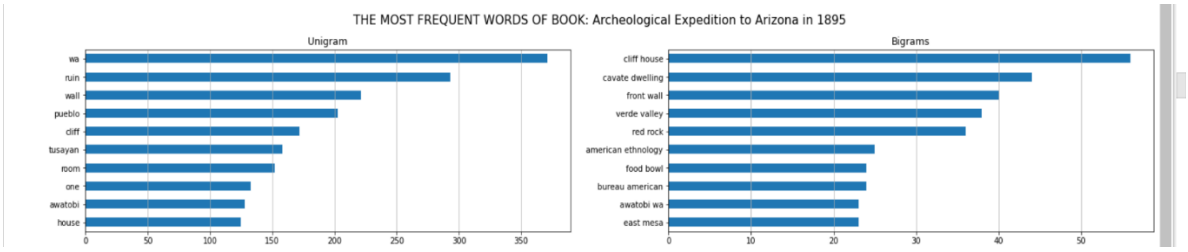
1- Unigram & Bigram

```
top= 10
for label in result['Title_of_Book'].unique():
    corpus = result[result["Title_of_Book"]==label]["PartitionsList"]
    lst_tokens = nltk.tokenize.word_tokenize(corpus.str.cat(sep=" "))
    fig, ax = plt.subplots(nrows=1, ncols=2,constrained_layout=True, figsize=(20, 4))
    fig.suptitle(f"THE MOST FREQUENT WORDS OF BOOK: {label} ", fontsize=15)

    #to draw the unigram gragh
    dic_words_freq = nltk.FreqDist(lst_tokens)
    result_unigram = pd.DataFrame(dic_words_freq.most_common(),
                        columns=["Word","Freq"])
    result_unigram.set_index("Word").iloc[:10,:].sort_values(by="Freq").plot(
                    kind="barh", title="Unigram", ax=ax[0],
                    legend=False).grid(axis='x')
    ax[0].set(ylabel=None)

    #to draw the bigram graph
    dic_words_freq = nltk.FreqDist(nltk.ngrams(lst_tokens, 2))
    result_bigram = pd.DataFrame(dic_words_freq.most_common(),
                        columns=["Word","Freq"])
    result_bigram["Word"] = result_bigram["Word"].apply(lambda x: " ".join(
                    string for string in x) )
    result_bigram.set_index("Word").iloc[:10,:].sort_values(by="Freq").plot(
                    kind="barh", title="Bigrams", ax=ax[1],
                    legend=False).grid(axis='x')
    ax[1].set(ylabel=None)
```

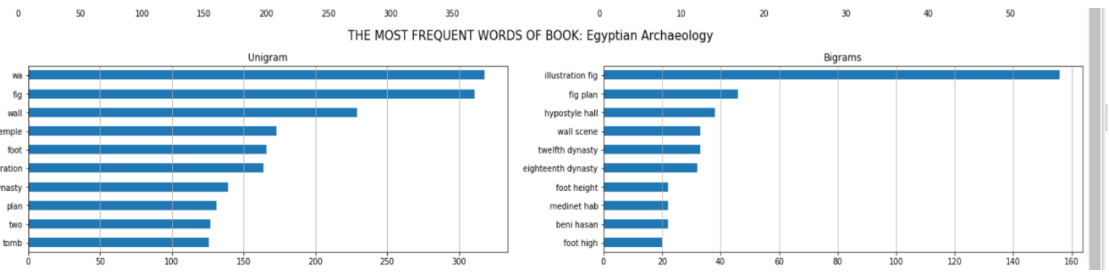
Pegan and Christian Rome book



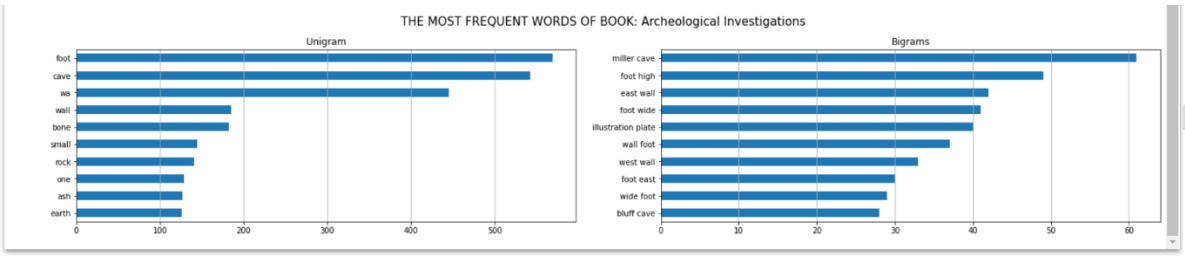
Archeological Expedition to Arizona in 1895 book



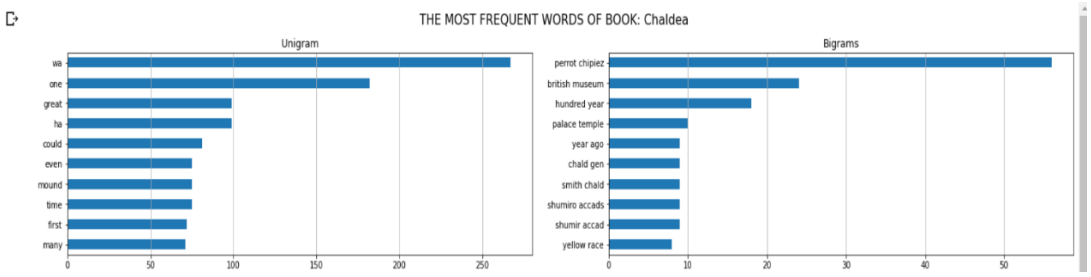
Egyptian Archaeology book



Archeological Investigation book



Chaldea book



2- Using WordCloud: to get the most frequent 50 words is every book.

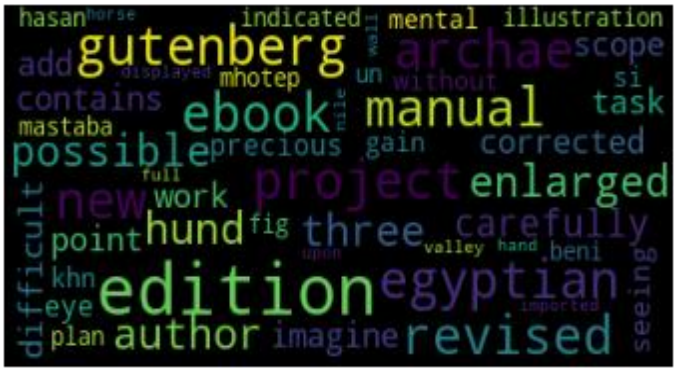
```
import wordcloud #Python wordcloud library to create tag clouds
import string
#for loop to take every unique book in the column of Title_of_Book
for n in result['Title_of_Book'].unique():
  books = result[result["Title_of_Book"]==n]["PartitionsList"]
 #to print the most frequent 50 words of the unique book
  print(f"\n THE MOST FREQUENT 50 WORDS OF BOOK CALLED: {n}\n")
  WordCloudGragh = wordcloud.WordCloud(background_color='black', max_words=50, max_font_size=40)
  WordCloudGragh = WordCloudGragh.generate(str(books))
  plt.axis('off')
  plt.imshow(WordCloudGragh, cmap=None)
  plt.show()
```

Chaldea book

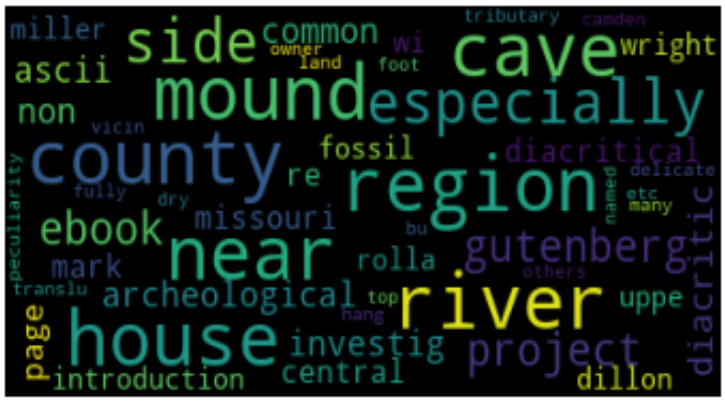
Pagan and Christian Rome

Archeological Expedition to Arizona in 1895

Egyptian Archaeology



Archeological Investigations



3- labeling the data: Important step for classification problems.

```
#to ensure that every book have 200 partition, and every partition have 100 words.
BooksWords = []  #to combine all words together
#for loop to get the book
for i in Books:
  l = i[0:(math.floor(len(i)/100)) * 100]
  BooksWords.append(l)
#to combine all lists of the words on a single dataframe
result = pd.DataFrame()
for i in range(len(BooksWords)):
    df = {}
    list_of_partitions =  [BooksWords[i][x:x+100] for x in range(0, len(BooksWords[i]), 100)]
    #to combine Book Authors in one column
    df['Author_of_Book']= BooksAuthors[i]
    #to combine Book Names in one column
    df['Title_of_Book']= BooksNames[i]
    #to combine Book Labels in one column
    df['Label_of_Book'] = BooksLabels[i]
    #to combine Book Partitions in one column
    df['PartitionsList'] = list_of_partitions
    data = pd.DataFrame(df)
    #for loop to join our data together
    for i in range(len(data)):
      data["PartitionsList"][i] = " ".join(data["PartitionsList"][i])
    final_result = data[:200]
    result = result.append(final_result)
```

```
#print head of data
result.head(5)
```

	Author_of_Book	Title_of_Book	Label_of_Book	PartitionsList
0	Zénaïde A. Ragozin	Chaldea	a	project gutenberg ebook chaldea ragozin ebook ...
1	Zénaïde A. Ragozin	Chaldea	a	study ancient history ragozin member soci ethn...
2	Zénaïde A. Ragozin	Chaldea	a	mesopotamia actual desolate state country plai...
3	Zénaïde A. Ragozin	Chaldea	a	every country culture art determined geographi...
4	Zénaïde A. Ragozin	Chaldea	a	destruction mound protection ruin contain refi...

4- Text transformation, Feature Engineering:

Text transformation techniques: BOW, TF-IDF, and N-Gram

1- BOW: A bag of words is a representation of text that describes the occurrence of words within a document.

	aahhotep	aal	abacus	abandon	abandoned	abandonment	abbey	abd	abdomen	abel	...	zigzag	zimmern	zip	zodiac	zodiacal	zone	zosimus	zowyet	zuguato	zur
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

1000 rows × 12168 columns

2- **TF-IDF:** Term frequency (TF) vectors show how important words are to documents. They are computed by using:

$$tf(term, document) = \frac{number\ of\ times\ the\ term\ occurs\ in\ the\ document}{total\ number\ of\ terms\ in\ the\ document}$$

	aahhotep	aal	abacus	abandon	abandoned	abandonment	abbey	abd	abdomen	abel	...	zigzag	zimmern	zip	zodiac	zodiacal	zone	zosimus	zowyet	zuguato	zur
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
996	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
997	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
998	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
999	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

1000 rows × 12168 columns

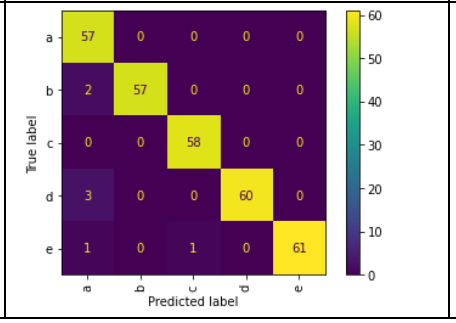
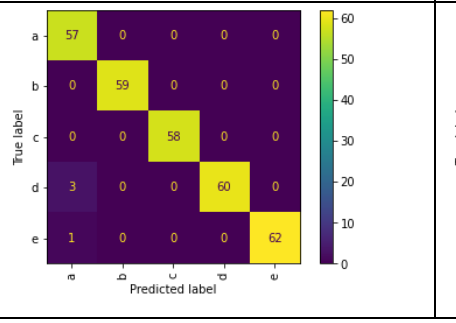
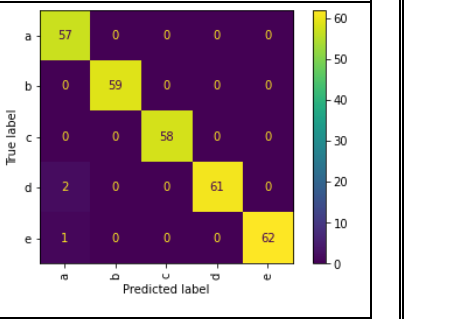
3- **N-Gram**: N-grams are continuous sequences of words or symbols or tokens in a document. In technical terms, they can be defined as the neighbouring sequences of items in a document.

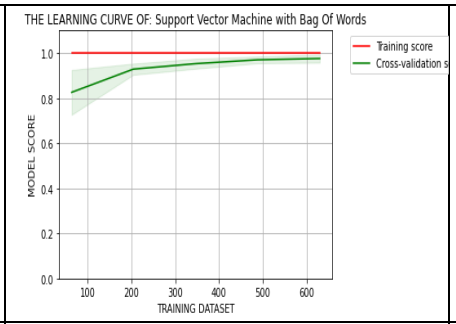
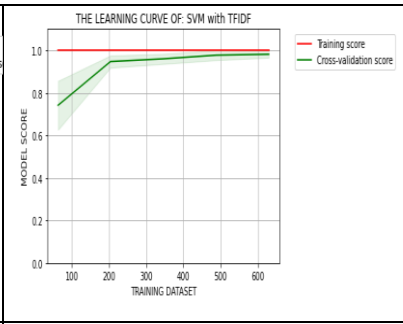
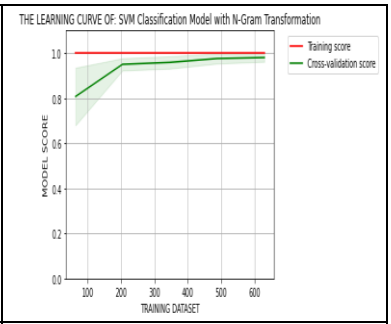
	aahhotep battle	aahhotep eighteenth	aahhotep funerary	aahhotep ring	aal wall	abacus carelessness	abacus fig	abacus hidden	abacus join	abandon congenial	...	zip chaldea	zip http	zodiac fashioned	zodiacal circle	zone pueblo	zosimus carmen	zowyet aryan	zuguato hinton	zur frage
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0
999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0

1000 rows × 80163 columns

5- Apply SVM, Decision Tree and KNN classification Models:

1- SVM classification model:

Model name	SVM with BOW	SVM with TF-IDF	SVM with N-Gram
Cross validation accuracy	97.5%	98.1%	98%
Testing accuracy	97.6%	98.6%	99%
Classification report	precision recall f1-score support a 0.90 1.00 0.95 57 b 1.00 0.97 0.98 59 c 0.98 1.00 0.99 58 d 1.00 0.95 0.98 63 e 1.00 0.97 0.98 63 accuracy 0.98 300 macro avg 0.98 0.98 0.98 300 weighted avg 0.98 0.98 0.98 300	precision recall f1-score support a 0.93 1.00 0.97 57 b 1.00 1.00 1.00 59 c 1.00 1.00 1.00 58 d 1.00 0.95 0.98 63 e 1.00 0.98 0.99 63 accuracy 0.99 300 macro avg 0.99 0.99 0.99 300 weighted avg 0.99 0.99 0.99 300	precision recall f1-score support a 0.95 1.00 0.97 57 b 1.00 1.00 1.00 59 c 1.00 1.00 1.00 58 d 1.00 0.97 0.98 63 e 1.00 0.98 0.99 63 accuracy 0.99 300 macro avg 0.99 0.99 0.99 300 weighted avg 0.99 0.99 0.99 300
Confusion matrix			

Learning curve			
Bias	0.030	0.030	0.030
Variance	0.005	0.008	0.005
Misclassification classes	7 classes are misclassified	4 classes are misclassified	3 classes are misclassified

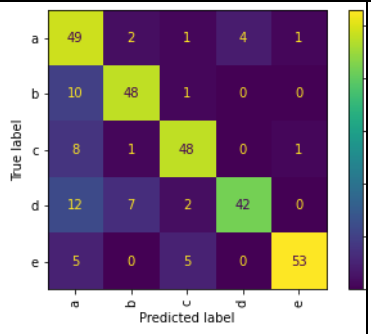
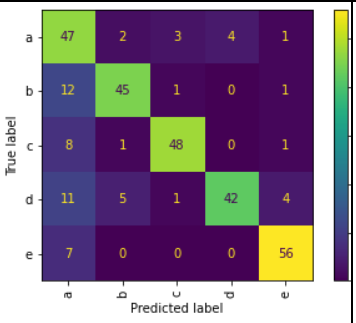
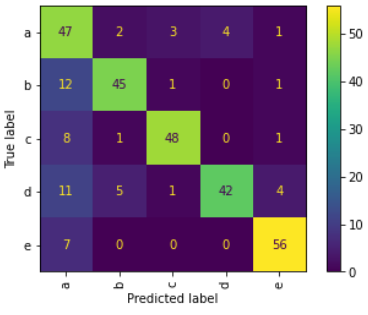
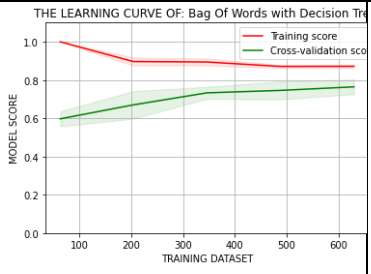
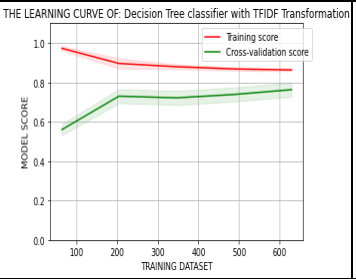
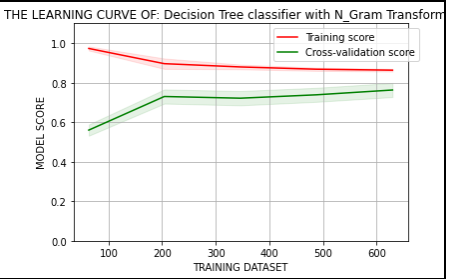
2- Decision Tree classification model

Model name	**DT with BOW**	**DT with TF-IDF**	**DT with N-Gram**
Cross validation accuracy	76.4%	76.2%	76.2%
Testing accuracy	80%	79.3%	79.3%
Classification report			

	precision	recall	f1-score	support
a	0.58	0.86	0.70	57
b	0.83	0.81	0.82	59
c	0.84	0.83	0.83	58
d	0.91	0.67	0.77	63
e	0.96	0.84	0.90	63
accuracy			0.80	300
macro avg	0.83	0.80	0.80	300
weighted avg	0.83	0.80	0.81	300

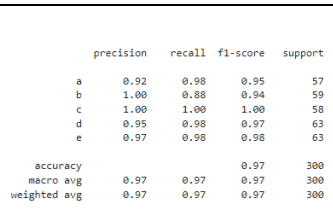
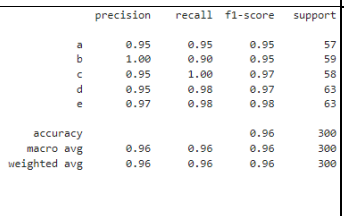
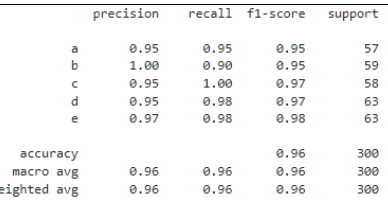
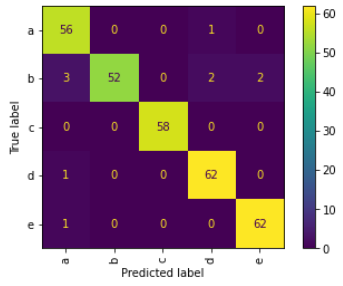
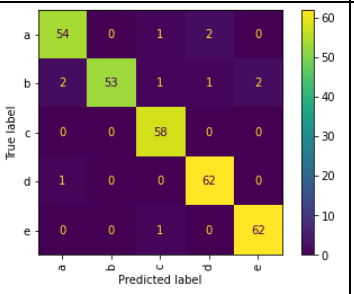
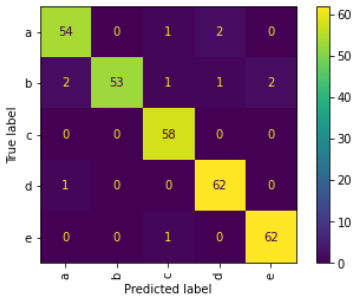
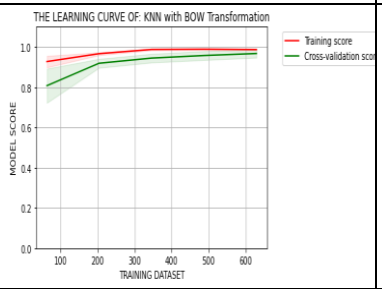
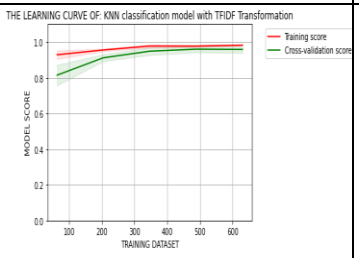
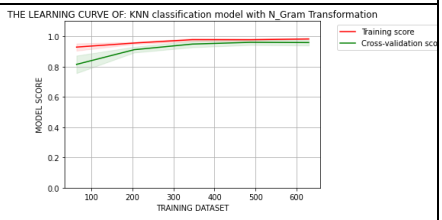
	precision	recall	f1-score	support
a	0.55	0.82	0.66	57
b	0.85	0.76	0.80	59
c	0.91	0.83	0.86	58
d	0.91	0.67	0.77	63
e	0.89	0.89	0.89	63
accuracy			0.79	300
macro avg	0.82	0.79	0.80	300
weighted avg	0.83	0.79	0.80	300

	precision	recall	f1-score	support
a	0.55	0.82	0.66	57
b	0.85	0.76	0.80	59
c	0.91	0.83	0.86	58
d	0.91	0.67	0.77	63
e	0.89	0.89	0.89	63
accuracy			0.79	300
macro avg	0.82	0.79	0.80	300
weighted avg	0.83	0.79	0.80	300

Confusion matrix			
Learning curve			
Bias	0.240	0.267	0.267
Variance	0.093	0.095	0.095

Misclassification classes	60 class are misclassified	62 class are misclassified	62 class are misclassified

3- KNN classification model

Model name	**KNN with BOW**	**KNN with TF-IDF**	**KNN with N-Gram**
Cross validation accuracy	96.7%	95.8%	95.8%
Testing accuracy	96.6%	96.3%	96.3%
Classification report			
Confusion matrix			
Learning curve			
Bias	0.067	0.063	0.063
Variance	0.030	0.038	0.038
Misclassification classes	Misclassified 10 classes	11 classes are misclassified	11 classes are misclassified

6- Champion Model:

After comparing our models together, the champion model is SVM with TF-IDF transformation.

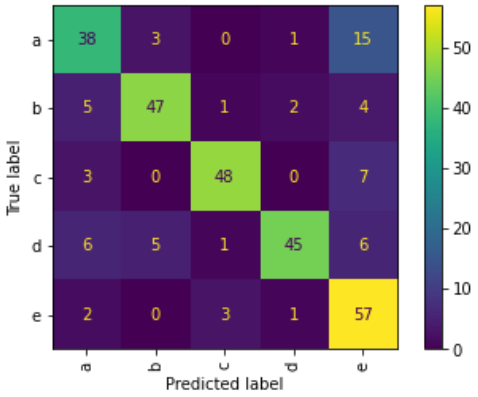
The model has an accuracy = 98.1%.

Change the kernel type to 'poly', and number of iterations are 3, so the cross validation accuracy is **71**% and testing accuracy is **78%**

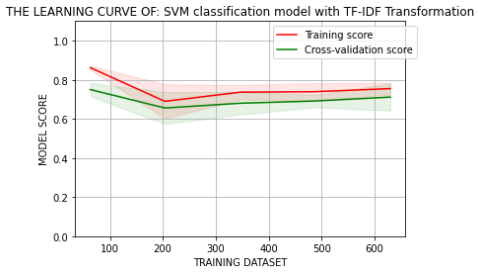
The classification report is:

	precision	recall	f1-score	support
a	0.70	0.67	0.68	57
b	0.85	0.80	0.82	59
c	0.91	0.83	0.86	58
d	0.92	0.71	0.80	63
e	0.64	0.90	0.75	63
accuracy			0.78	300
macro avg	0.80	0.78	0.79	300
weighted avg	0.80	0.78	0.79	300

The confusion matrix:



The learning curve:



7 -cross validation:

The k-fold cross-validation procedure is a standard method for estimating the performance of a machine learning algorithm on a dataset.

When applying K-fold cross validation on SVM TF-IDF model, and k = 10, the accuracy matrix is:

[1. 0.97142857 0.98571429 0.98571429 0.95714286 1.

0.95714286 0.98571429 1. 0.97142857]

8- perform Error Analysis:

When performing the error analysis of the champion model, the misclassified classes are 4.

[93] TFIDF_SVM_Error

	Error	Right Label	Predicted Label
0	project gutenberg ebook archeological investig...	e	a
1	task would difficult imagine point work scope ...	d	a
2	day even whole day brick piled stack wise air ...	d	a
3	new edition revised enlarged author three hund...	d	a

THE AVERAGE BIAS IS: 0.030

THE AVEARAGE VARIANCE IS: 0.008

So we made some analysis to discover a pattern for misclassified classes using wordCloud, the model has a conflict when classifying label a and d together.

[project gutenberg ebook archeological investigation gerard fowke ebook use anyone anywhere cost almost restriction whatsoever may copy give away use term project gutenberg license included ebook online www gutenberg org title archeological investigation bureau american ethnology bulletin author gerard fowke release date july ebook language english character set encoding iso start project gutenberg ebook archeological investigation produced robert connal jeannie howse alicia williams bruce albrecht online distributed proofreading team http www pgdp net file wa produced image generously made available biblioth que nationale france bnf gallica http gallica bnf transcriber note inconsistent hyphenation match original document text contains character]

[task would difficult imagine point work scope character better calculated give lasting delight class reader skilled archaeologist page contain new fact new view new interpretation know little perhaps nothing subject discussion open fresh fascinating field study enough say handbook egyptian archaeology wa much needed professor maspero ha given exactly required ha done much ha given picturesque vivacious highly original volume delightful learned instructive dull regard practical side archaeology ought unnecessary point usefulness strictly parallel usefulness public museum collect exhibit object ancient art industry worse idle also endeavour disseminate knowledge history art industry process employed artist craftsman past archaeology le love]

[day even whole day brick piled stack wise air circulate freely among remain week two used frequently however exposed hour heat sun building begun yet damp mud however tenacious notwithstanding carelessness readily put shape outer face brick become disintegrated action weather inner part wall remain intact still separable good modern workman easily mould thousand brick day week practice may turn even ancient workman whose appliance wise differed present day produced equally satisfactory result dimension generally adopted inch ordinary brick larger size note though larger smaller often met ruin brick issued royal workshop sometimes stamped cartouch reigning monarch made private factory]

[new edition revised enlarged author three hundred nine illustration preface fourth revised edition notwithstanding fact egyptology recognised science exact communicable knowledge whose existence scope behoves modern culture take cognisance work maspero still remains handbook egyptian archaeology egyptology yet infancy whatever age egyptologist long die young every year almost every month fresh material study found fresh light thrown upon progress excavation exploration research hence follows course year standard text book require considerable addition modification greatest value student must always start foremost vantage ground increasing demand egyptian archaeology english american tourist well student decided english publisher issue new edition light portable form]

- The model has a conflict with these paragraphs.

9. Conclusion

We have learned many new things during this assignment, and we have discovered some useful techniques. We have gotten familiar with new libraries.

We have learnt how to use the transformation and apply it in each model and evaluate the accuracy and confusion matrix to each model. Now, we can say that we are capable of dealing with SVM , KNN and Decision Tree models.

Also apply K- fold Cross validation and how to choose the champion model and make Error analysis

10- References:

1- https://scikit-learn.org/stable/modules/cross_validation.html
2- https://pythonwife.com/the-support-vector-machines-svm-algorithm-for-nlp/
3- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.learning_curve.html