In [21]:

```python
# Symon Kimitei
# Neural Nets and Deep Learning
# Term: Spring 2021
# Program: Sentence Classification using the KNN Algorithm
# Due Date: 2/21/2021
#=======================================================================
# importing the required dependencies
import os
import math
import nltk
import itertools
#nltk.download('stopwords')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import Pipeline, FeatureUnion, make_pipeline
from sklearn.preprocessing import OneHotEncoder
from sklearn.preprocessing import LabelEncoder
from sklearn import preprocessing
%matplotlib inline
from nltk.corpus import stopwords
stopWords = set(stopwords.words('english'))
# nltk.download('punkt')
from nltk import word_tokenize

# Specify the working directory
os.chdir("C:/Users/kimit/OneDrive/Desktop/py")
df = pd.read_csv("train_set.csv")
```

In [23]:

```
# Tokenize every word in each sentence on each row of the Train dataset datafr
ame
# nnd display the first 10 rows
df['word_tokens'] = df["Words (split by space)"].apply(word_tokenize)
df.head(10)
```

Out[23]:

	Words (split by space)	label	word_tokens
0	europe retain trophy with big win	joy	[europe, retain, trophy, with, big, win]
1	senate votes to revoke pensions	sad	[senate, votes, to, revoke, pensions]
2	the amounts you have to pay for a bomb scare	fear	[the, amounts, you, have, to, pay, for, a, bom...
3	pair of satellites will document sun in d	joy	[pair, of, satellites, will, document, sun, in...
4	malaysian airasia x to fly in july	joy	[malaysian, airasia, x, to, fly, in, july]
5	dow hits new record eyes	joy	[dow, hits, new, record, eyes]
6	bathing mom awakes to find baby dead	sad	[bathing, mom, awakes, to, find, baby, dead]
7	we re a pretty kind bully	joy	[we, re, a, pretty, kind, bully]
8	women in their s are perfectly good mothers	sad	[women, in, their, s, are, perfectly, good, mo...
9	hands on doomsday clock move forward	fear	[hands, on, doomsday, clock, move, forward]

In [25]:

```
# Save only the unique tokenized words from the Word_tokens column of the trai
n dataset
# and print the first 10 words, sorted in ascending order
# unique_words = list(set(itertools.chain.from_iterable(df['word_tokens'])))
# Also Remove stop words such as (an, a, the, etc.)
unique_words=[word for word in set(itertools.chain.from_iterable(df['word_toke
ns']))
              if word not in stopWords]
unique_words.sort()
unique_words[:10]
```

Out[25]:

```
['abbas',
 'abduct',
 'abducting',
 'abductor',
 'abdul',
 'abilities',
 'abortions',
 'abuse',
 'accessories',
 'accidentally']
```

In [26]:

```
# Create a function Sentence_one_hot that takes as input the unique words,
# and checks whether a word is in the word tokens or not
# Returns a Boolean, 0 or 1.
def sentences_one_hot(word_tokens, unique_words=unique_words):
    return [int(word in word_tokens) for word in unique_words]
```

In [28]:

```
# Create an encoded list of booleans for each sentence by using the Sentences_
one_hot function.
# 1 means the word exists in a row and 0 otherwise
one_hot_words = pd.DataFrame(df['word_tokens'].apply(sentences_one_hot).values
.tolist(),
                             columns=unique_words)
one_hot_words.head(10)
```

Out[28]:

	abbas	abduct	abducting	abductor	abdul	abilities	abortions	abuse	accessories	accidenta
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	0	
8	0	0	0	0	0	0	0	0	0	
9	0	0	0	0	0	0	0	0	0	

10 rows × 1938 columns

In [30]:

```
# Tokenize the sentences in the test_set and apply one hot encoding to them
test_df = pd.read_csv("test_set.csv")
test_df['word_tokens'] = test_df["Words (split by space)"].apply(word_tokenize
)
test_one_hot_words = pd.DataFrame(test_df['word_tokens'].apply(sentences_one_h
ot).values.tolist(),
                                  columns=unique_words)
# Display the first 5 rows of the one hot encoded test dataset
test_one_hot_words.head(5)
```

Out[30]:

	abbas	abduct	abducting	abductor	abdul	abilities	abortions	abuse	accessories	accidenta
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	

5 rows × 1938 columns

In [31]:

```
# Tokenize the sentences in the validation_set and apply one hot encoding to t
hem
val_df = pd.read_csv("validation_set.csv")
val_df['word_tokens'] = val_df["Words (split by space)"].apply(word_tokenize)
val_one_hot_words = pd.DataFrame(val_df['word_tokens'].apply(sentences_one_hot
).values.tolist(),
                                 columns=unique_words)
# Display the first 5 rows of the one hot encoded validation dataset
val_one_hot_words.head(5)
```

Out[31]:

	abbas	abduct	abducting	abductor	abdul	abilities	abortions	abuse	accessories	accidenta
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	

5 rows × 1938 columns

In [13]:

```
# Find the number of rows in the validation set
index=val_df.index
N=len(index)
N=math.sqrt(N)
N
```

Out[13]:

```
17.635192088548397
```

In [32]:

```
# Calculate the training set and the validation set accuracies and store them
 in a dataframe
# Loop through for each k value where the maximum k used is N which is the squ
are root of
# the maximum number of rows in the validation dataset .
# Display a table of the output of the train and validation accuracies for the
models
#-----------------------------------------------------------------------------
--------
# 'k' in KNN is a parameter that refers to the number of nearest neighbours to
include
# in the majority of the voting process.

best_k=1
bestk_valid_acc=0

KNN= pd.DataFrame({'k': [],'train_accuracy': [],'validation_accuracy': []})
for k in range(3,int(N)+1,2):   # range(3,int(N)+1,2) = 3,5,7,9,....N
    # when p = 1, we're using the manhattan distance (euclidean: p = 2)
    classifier = KNeighborsClassifier(n_neighbors=k,p=1)
    classifier.fit(one_hot_words,df.label.values)

    # Calculate the accuracy for the training data set for a given k - value
    train_acc=np.mean(classifier.predict(one_hot_words) == df.label)

     # Calculate the accuracy for the validation data set for a given k-value
    valid_acc= np.mean(classifier.predict(val_one_hot_words) == val_df.label)

    # Search for the best k for the validation set. Best k yields the highest
 accuracy
    if (valid_acc>bestk_valid_acc):
        bestk_valid_acc=valid_acc
        best_k=k
    KNN = KNN.append({'k':k,'train_accuracy':train_acc,'validation_accuracy':v
alid_acc},ignore_index=True)

KNN
```

Out[32]:

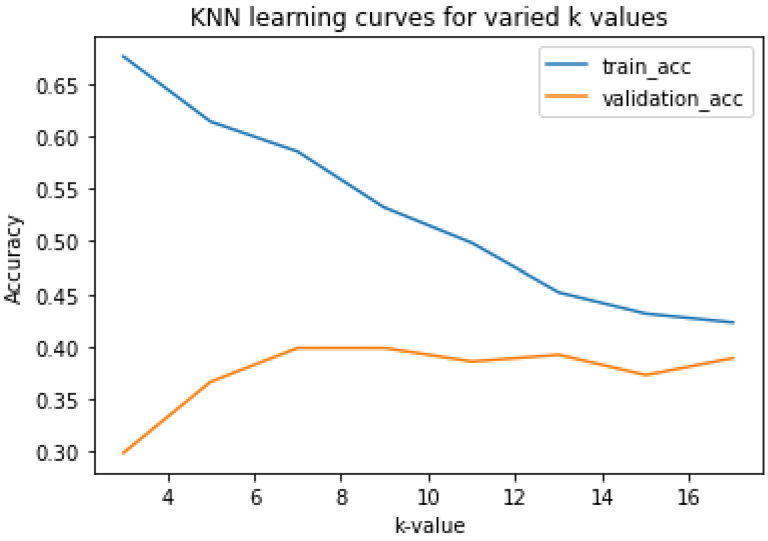
	k	train_accuracy	validation_accuracy
0	3.0	0.675585	0.299035
1	5.0	0.613712	0.366559
2	7.0	0.585284	0.398714
3	9.0	0.531773	0.398714
4	11.0	0.498328	0.385852
5	13.0	0.451505	0.392283
6	15.0	0.431438	0.372990
7	17.0	0.423077	0.389068

In [33]:

```
# saving the results dataframe in Ms Excel
KNN.to_csv('KNN.csv', header=True, index=False)
```

In [34]:

```
# Display a graph of the accuracy for the train and validation data sets vs k-
value
lines = KNN.plot.line(x='k', y=['train_accuracy', 'validation_accuracy'])
plt.title('KNN learning curves for varied k values')
plt.legend(['train_acc', 'validation_acc'], loc='upper right')
plt.xlabel('k-value')
plt.ylabel('Accuracy')
plt.show()
```



In [35]:

```
# Print the optimal k value for the validation set
# The k value is used in the test set to predict labels
print("Best k value=",best_k)

# Fit the KNN model to the test dataset using the optimal k value
# when p = 1, we are using the manhattan distance (euclidean: p = 2)
classifier = KNeighborsClassifier(n_neighbors=best_k,p=1)
classifier.fit(one_hot_words,df.label.values)
# Predict the label for each sentence in the test dataset
test_df['label'] = classifier.predict(test_one_hot_words)

test_df=test_df[['textid', 'Words (split by space)', 'label']]
test_df.head(10)
```

```
Best k value= 7
```

Out[35]:

	textid	Words (split by space)	label
0	1	senator carl krueger thinks ipods can kill you	joy
1	2	who is prince frederic von anhalt	joy
2	3	prestige has magic touch	surprise
3	4	study female seals picky about mates	joy
4	5	no e book for harry potter vii	joy
5	6	blair apologises over friendly fire inquest	fear
6	7	vegetables may boost brain power in older adults	surprise
7	8	afghan forces retake town that was overrun by ...	sad
8	9	skip the showers male sweat turns women on stu...	surprise
9	10	made in china irks some burberry shoppers	joy

In [36]:

```
# Save the test results in Ms Excel
test_df.to_csv('Test_results.csv', header=True, index=False)
```