In [1]:
```
!pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\admin\anaconda3\lib\site-packages
(3.6.5)
Requirement already satisfied: click in c:\users\admin\anaconda3\lib\site-packages
(from nltk) (8.0.3)
Requirement already satisfied: joblib in c:\users\admin\anaconda3\lib\site-packages
(from nltk) (1.1.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\admin\anaconda3\lib\site-
packages (from nltk) (2021.8.3)
Requirement already satisfied: tqdm in c:\users\admin\anaconda3\lib\site-packages (f
rom nltk) (4.62.3)
Requirement already satisfied: colorama in c:\users\admin\anaconda3\lib\site-package
s (from click->nltk) (0.4.4)
```

In [3]:
```
import nltk
#nltk.download_shell()
```

In [2]:
```
messages = [line.rstrip() for line in open(r'C:\Users\ADMIN\Desktop\Projects\Data se
print(len(messages))
```

```
4846
```

In [5]:
```
for message_no, message in enumerate(messages[:10]):
    print(message_no, message)
    print('\n')
```

```
0 neutral,"According to Gran , the company has no plans to move all production to Ru
ssia , although that is where the company is growing ."


1 neutral,"Technopolis plans to develop in stages an area of no less than 100,000 sq
uare meters in order to host companies working in computer technologies and telecomm
unications , the statement said ."


2 negative,"The international electronic industry company Elcoteq has laid off tens
of employees from its Tallinn facility ; contrary to earlier layoffs the company con
tracted the ranks of its office workers , the daily Postimees reported ."


3 positive,With the new production plant the company would increase its capacity to
meet the expected increase in demand and would improve the use of raw materials and
therefore increase the production profitability .


4 positive,"According to the company 's updated strategy for the years 2009-2012 , B
asware targets a long-term net sales growth in the range of 20 % -40 % with an opera
ting profit margin of 10 % -20 % of net sales ."


5 positive,FINANCING OF ASPOCOMP 'S GROWTH Aspocomp is aggressively pursuing its gro
wth strategy by increasingly focusing on technologically more demanding HDI printed
circuit boards PCBs .


6 positive,"For the last quarter of 2010 , Componenta 's net sales doubled to EUR131
m from EUR76m for the same period a year earlier , while it moved to a zero pre-tax
profit from a pre-tax loss of EUR7m ."
```

```
7 positive,"In the third quarter of 2010 , net sales increased by 5.2 % to EUR 205.5
mn , and operating profit by 34.9 % to EUR 23.5 mn ."


8 positive,Operating profit rose to EUR 13.1 mn from EUR 8.7 mn in the corresponding
period in 2007 representing 7.7 % of net sales .


9 positive,"Operating profit totalled EUR 21.1 mn , up from EUR 18.6 mn in 2007 , re
presenting 9.7 % of net sales ."
```

In [6]:
```
import pandas as pd
messages = pd.read_csv(r'C:\Users\ADMIN\Desktop\Projects\Data sets\FinancialData.csv
messages.head()
```

Out[6]:

	label	message
0	neutral	According to Gran , the company has no plans t...
1	neutral	Technopolis plans to develop in stages an area...
2	negative	The international electronic industry company ...
3	positive	With the new production plant the company woul...
4	positive	According to the company 's updated strategy f...

In [8]:
```
messages.describe()
```

Out[8]:

	label	message
count	4846	4846
unique	3	4838
top	neutral	TELECOMWORLDWIRE-7 April 2006-TJ Group Plc sel...
freq	2879	2

In [9]:
```
messages.groupby('label').describe()
```

Out[9]:

	message			
	count	**unique**	**top**	**freq**
label				
negative	604	604	The international electronic industry company ...	1
neutral	2879	2873	SSH Communications Security Corporation is hea...	2
positive	1363	1363	With the new production plant the company woul...	1

In [10]:
```
messages['length'] = messages['message'].apply(len)
messages.head()
```

Out[10]:

	label	message	length
0	neutral	According to Gran , the company has no plans t...	127
1	neutral	Technopolis plans to develop in stages an area...	190
2	negative	The international electronic industry company ...	228
3	positive	With the new production plant the company woul...	206
4	positive	According to the company 's updated strategy f...	203

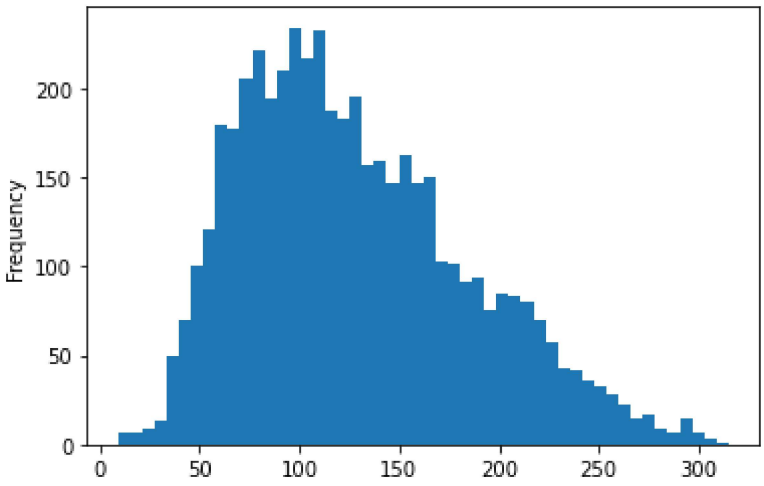
In [11]:

```
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
messages['length'].plot(bins=50, kind='hist')
```

Out[11]:

```
<AxesSubplot:ylabel='Frequency'>
```



In [12]:

```
messages.length.describe()
```

Out[12]:

```
count    4846.000000
mean      128.132068
std        56.526180
min         9.000000
25%        84.000000
50%       119.000000
75%       163.000000
max       315.000000
Name: length, dtype: float64
```

In [13]:

```
messages[messages['length'] == 315]['message'].iloc[0]
```

Out[13]:

```
'Supported Nokia phones include : N96 , N95-8GB , N95 , N93-N931 , N92 , N85 , N82 ,
N81 , N80 , N79 , N78 , N77 , N76 , N75 , N73 , N72 , N71 , E90 , E71 , E70 , E66 ,
E65 , E62 , E61-E61i , E60 , E51 , E50 , Touch Xpress 5800 , 6220 Classic , 6210 Nav
igator , 6120 Classic , 6110 Navigator , 5700 , 5500 , 5320XM .'
```

Text Pre-processing and Data Cleaning

In [15]:

```
messages.head()
```

Out[15]:

	label	message	length
0	neutral	According to Gran , the company has no plans t...	127
1	neutral	Technopolis plans to develop in stages an area...	190
2	negative	The international electronic industry company ...	228
3	positive	With the new production plant the company woul...	206
4	positive	According to the company 's updated strategy f...	203

Now let's "tokenize" these messages. Tokenization is just the term used to describe the process of converting the normal text strings in to a list of tokens (words that we actually want).

In [17]:

```
import string
from nltk.corpus import stopwords
def text_process(mess):
    """
    Takes in a string of text, then performs the following:
    1. Remove all punctuation
    2. Remove all stopwords
    3. Returns a list of the cleaned text
    """
    # Check characters to see if they are in punctuation
    nopunc = [char for char in mess if char not in string.punctuation]

    # Join the characters again to form the string.
    nopunc = ''.join(nopunc)

    # Now just remove any stopwords
    return [word for word in nopunc.split() if word.lower() not in stopwords.words('
messages['message'].head(5).apply(text_process)
```

Out[17]:

```
0    [According, Gran, company, plans, move, produc...
1    [Technopolis, plans, develop, stages, area, le...
2    [international, electronic, industry, company,...
3    [new, production, plant, company, would, incre...
4    [According, company, updated, strategy, years,...
Name: message, dtype: object
```

Vectorization

In [18]:

```
from sklearn.feature_extraction.text import CountVectorizer
bow_transformer = CountVectorizer(analyzer=text_process).fit(messages['message'])

# Print total number of vocab words
print(len(bow_transformer.vocabulary_))
```

```
12278
```

In [19]:

```
message4 = messages['message'][3]
print(message4)
```

```
With the new production plant the company would increase its capacity to meet the ex
pected increase in demand and would improve the use of raw materials and therefore i
ncrease the production profitability .
```

In [20]:

```
bow4 = bow_transformer.transform([message4])
print(bow4)
```

```
print(bow4.shape)
```

```
  (0, 6987)     1
  (0, 7255)     1
  (0, 7639)     1
  (0, 8154)     1
  (0, 8848)     1
  (0, 8877)     3
  (0, 9446)     1
  (0, 9485)     1
  (0, 9721)     1
  (0, 10143)    1
  (0, 10340)    2
  (0, 10352)    1
  (0, 10503)    1
  (0, 11660)    1
  (0, 11952)    1
  (0, 12206)    2
(1, 12278)
```

In [21]:

```
print(bow_transformer.get_feature_names()[6987])
print(bow_transformer.get_feature_names()[12206])
```

```
capacity
would
```

In [22]:

```
messages_bow = bow_transformer.transform(messages['message'])
```

In [23]:

```
print('Shape of Sparse Matrix: ', messages_bow.shape)
print('Amount of Non-Zero occurences: ', messages_bow.nnz)
```

```
Shape of Sparse Matrix:  (4846, 12278)
Amount of Non-Zero occurences:  60652
```

In [24]:

```
sparsity = (100.0 * messages_bow.nnz / (messages_bow.shape[0] * messages_bow.shape[1
print('sparsity: {}'.format(round(sparsity)))
```

```
sparsity: 0
```

In [25]:

```
from sklearn.feature_extraction.text import TfidfTransformer

tfidf_transformer = TfidfTransformer().fit(messages_bow)
tfidf4 = tfidf_transformer.transform(bow4)
print(tfidf4)
```

```
  (0, 12206)    0.34006531984231525
  (0, 11952)    0.18401976136204104
  (0, 11660)    0.261790357183673
  (0, 10503)    0.21916674125837723
  (0, 10352)    0.24441018572286424
  (0, 10340)    0.3226250661727878
  (0, 10143)    0.16221188996815172
  (0, 9721)     0.13728819203806375
  (0, 9485)     0.2479949389970483
  (0, 9446)     0.21061609765147357
  (0, 8877)     0.5002653610822685
  (0, 8848)     0.21607154589277733
  (0, 8154)     0.17452132143656565
  (0, 7639)     0.21323459607543088
```

```
  (0, 7255)     0.09710746479542764
  (0, 6987)     0.1917564023250975
```

In [26]:
```
messages_tfidf = tfidf_transformer.transform(messages_bow)
print(messages_tfidf.shape)
```

```
(4846, 12278)
```

Training a model

In [28]:
```
from sklearn.naive_bayes import MultinomialNB
spam_detect_model = MultinomialNB().fit(messages_tfidf, messages['label'])
```

In [29]:
```
print('predicted:', spam_detect_model.predict(tfidf4)[0])
print('expected:', messages.label[1])
```

```
predicted: neutral
expected: neutral
```

Test the data

In [30]:
```
all_predictions = spam_detect_model.predict(messages_tfidf)
print(all_predictions)
```

```
['neutral' 'neutral' 'neutral' ... 'positive' 'positive' 'neutral']
```

In [31]:
```
from sklearn.metrics import classification_report
print (classification_report(messages['label'], all_predictions))
```

```
              precision    recall  f1-score   support

    negative       1.00      0.10      0.18       604
     neutral       0.74      0.99      0.85      2879
    positive       0.76      0.52      0.62      1363

    accuracy                           0.75      4846
   macro avg       0.83      0.54      0.55      4846
weighted avg       0.78      0.75      0.70      4846
```

In [32]:
```
from sklearn.model_selection import train_test_split

msg_train, msg_test, label_train, label_test = \
train_test_split(messages['message'], messages['label'], test_size=0.2)

print(len(msg_train), len(msg_test), len(msg_train) + len(msg_test))
```

```
3876 970 4846
```

In [35]:
```
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('bow', CountVectorizer(analyzer=text_process)),  # strings to token integer cou
    ('tfidf', TfidfTransformer()),  # integer counts to weighted TF-IDF scores
    ('classifier', MultinomialNB()),  # train on TF-IDF vectors w/ Naive Bayes class
])
```

In [34]:
```
pipeline.fit(msg_train,label_train)
```

Out[34]:
```
Pipeline(steps=[('bow',
                 CountVectorizer(analyzer=<function text_process at 0x000001BABCB566
70>)),
                ('tfidf', TfidfTransformer()),
                ('classifier', MultinomialNB())])
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: