In [1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```python
spam_df = pd.read_csv(r"C:\Users\Owner\Downloads\spam.csv",encoding='latin1')
spam_df
```

Out[2]:

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN
...	...	...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	NaN	NaN	NaN
5568	ham	Will Ì_ b going to esplanade fr home?	NaN	NaN	NaN
5569	ham	Pity, * was in mood for that. So...any other s...	NaN	NaN	NaN
5570	ham	The guy did some bitching but I acted like i'd...	NaN	NaN	NaN
5571	ham	Rofl. Its true to its name	NaN	NaN	NaN

5572 rows × 5 columns

In [3]:
```python
spam_df.shape
```

Out[3]:
```
(5572, 5)
```

DATA CLEANING

In [4]:
```python
spam_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 5 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   v1          5572 non-null   object
 1   v2          5572 non-null   object
 2   Unnamed: 2  50 non-null     object
 3   Unnamed: 3  12 non-null     object
 4   Unnamed: 4  6 non-null      object
dtypes: object(5)
memory usage: 217.8+ KB
```

In [5]:
```python
# Dropping unnessary columns
spam_df.drop(columns=['Unnamed: 2','Unnamed: 3','Unnamed: 4'],inplace=True)
```

In [6]:
```python
spam_df.head()
```

Out[6]:

	v1	v2
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

In [7]:
```python
# Changing column names
spam_df.rename(columns={'v1':'Target', 'v2':'Text'},inplace=True)
```

In [8]:
```python
spam_df.head()
```

Out[8]:

	Target	Text
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

In [9]:
```python
# Checking for null values
spam_df.isna().sum()
```

Out[9]:
```
Target    0
Text      0
dtype: int64
```

In [10]:
```python
# Checking Duplicate values
spam_df.duplicated().sum()
```

Out[10]:
```
403
```

In [11]:
```python
spam_df.describe()
```

Out[11]:

	Target	Text
count	5572	5572
unique	2	5169
top	ham	Sorry, I'll call later
freq	4825	30

VISUALIZE DATASET

In [12]:
```python
# Let's get the length of the messages
spam_df['length'] = spam_df['Text'].apply(len)
spam_df.head()
```

Out[12]:

	Target	Text	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49
4	ham	Nah I don't think he goes to usf, he lives aro...	61

In [13]:
```python
spam_df
```

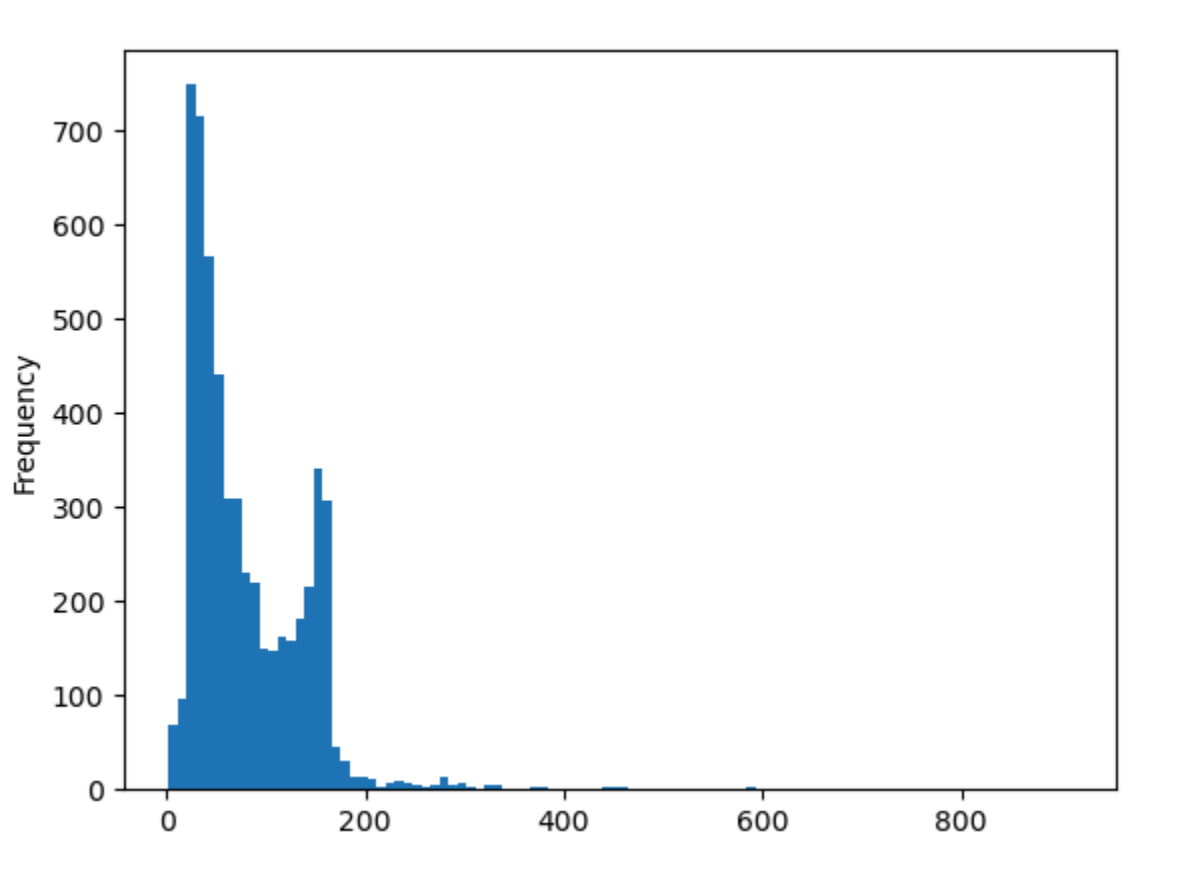
Out[13]:

	Target	Text	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49
4	ham	Nah I don't think he goes to usf, he lives aro...	61
...	...	...	...
5567	spam	This is the 2nd time we have tried 2 contact u...	161
5568	ham	Will Ì_ b going to esplanade fr home?	37
5569	ham	Pity, * was in mood for that. So...any other s...	57
5570	ham	The guy did some bitching but I acted like i'd...	125
5571	ham	Rofl. Its true to its name	26

5572 rows × 3 columns

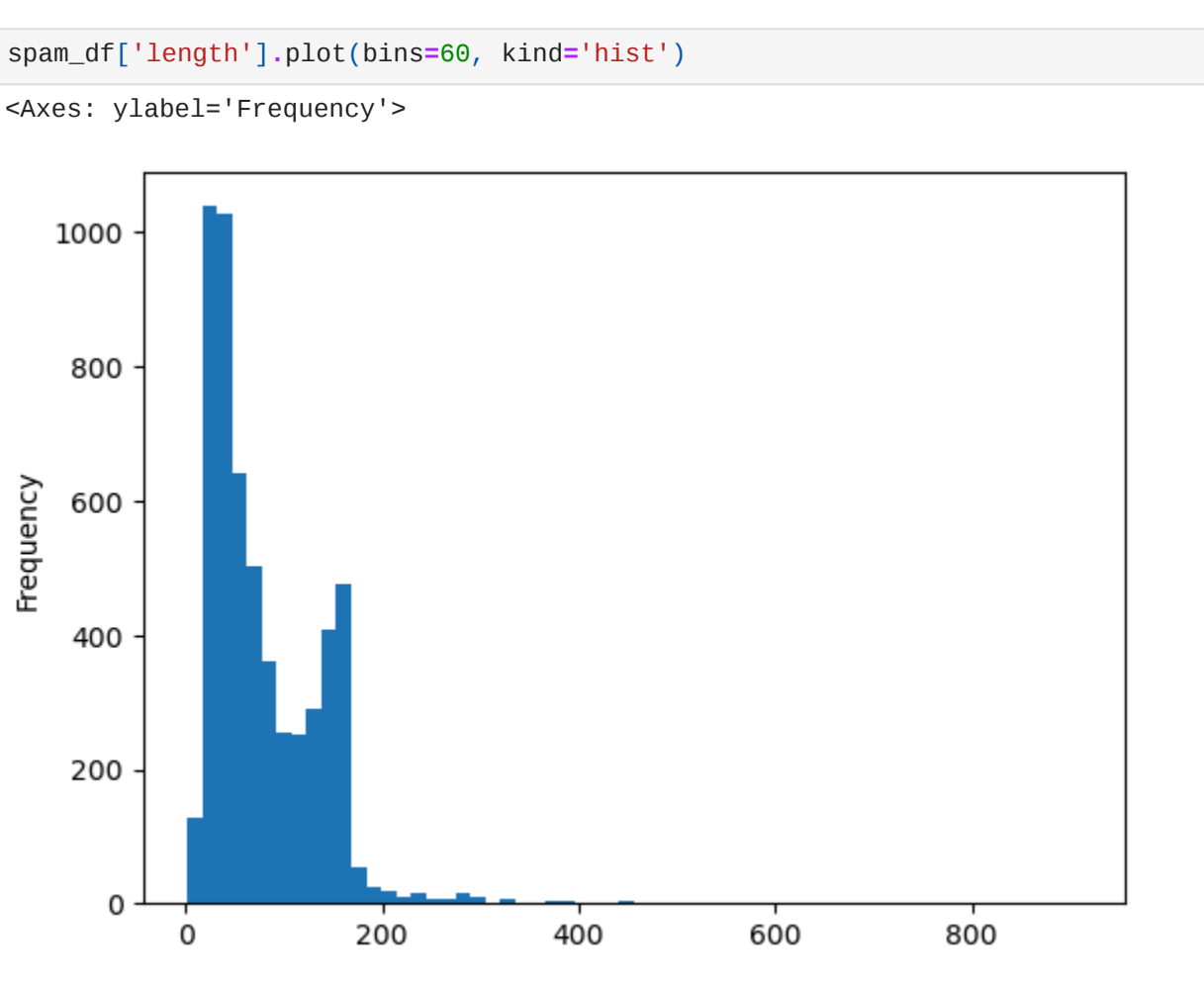
In [14]:
```python
spam_df['length'].plot(bins=100, kind='hist')
```

Out[14]:
```
<Axes: ylabel='Frequency'>
```



In [15]:
```python
spam_df['length'].plot(bins=60, kind='hist')
```

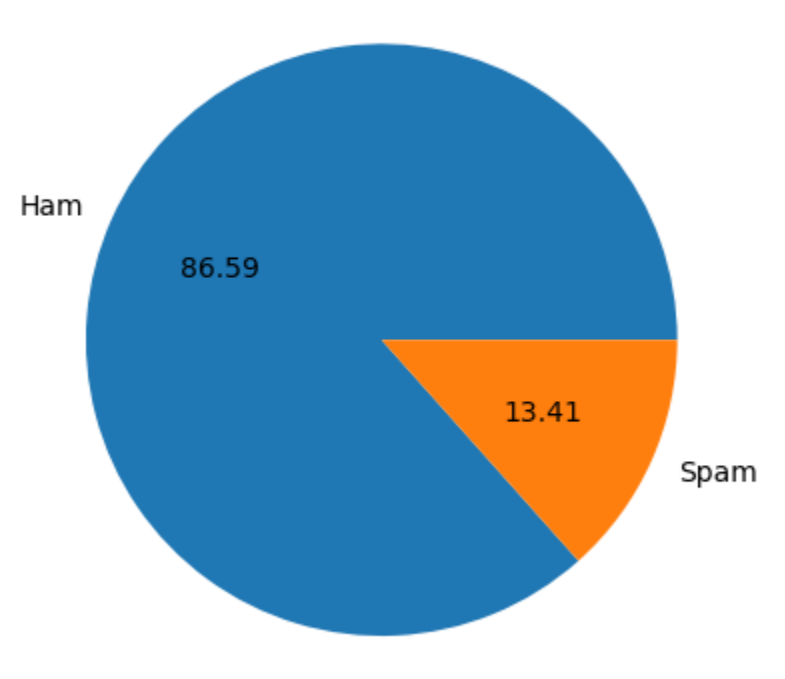
Out[15]:
```
<Axes: ylabel='Frequency'>
```



In [16]:
```python
# Ham and Spam count
spam_df['Target'].value_counts()
```

Out[16]:
```
ham     4825
spam     747
Name: Target, dtype: int64
```

In [17]:
```python
# Check the Target value using pie chart
plt.pie(spam_df['Target'].value_counts(), labels=['Ham','Spam'],autopct='%.2f')
plt.show()
```



In [18]:
```python
import nltk
nltk.download('punkt')
from nltk.corpus import stopwords
nltk.download('stopwords')
import string
string.punctuation
from nltk.stem.porter import PorterStemmer
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Owner\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Owner\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [ ]:

CREATE TESTING AND TRAINING DATASET/DATA CLEANING

In [22]:
```python
import string
string.punctuation
```

Out[22]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [23]:
```python
Test = 'Hello Mr. Future, I am so happy to be learning AI now!!'
```

In [24]:
```python
Test_punc_removed = [char for char in Test if char not in string.punctuation]
Test_punc_removed
```

Out[24]:
```
['H',
 'e',
 'l',
 'l',
 'o',
 ' ',
 'M',
 'r',
 ' ',
 'F',
 'u',
 't',
 'u',
 'r',
 'e',
 ' ',
 'I',
 ' ',
 'a',
 'm',
 ' ',
 's',
 'o',
 ' ',
 'h',
 'a',
 'p',
 'p',
 'y',
 ' ',
 't',
 'o',
 ' ',
 'b',
 'e',
 ' ',
 'l',
 'e',
 'a',
 'r',
 'n',
 'i',
 'n',
 'g',
 ' ',
 'A',
 'I',
 ' ',
 'n',
 'o',
 'w']
```

In [25]:
```python
# Join the characters again to form the string.
Test_punc_removed_join = ''.join(Test_punc_removed)
Test_punc_removed_join
```

Out[25]:
```
'Hello Mr Future I am so happy to be learning AI now'
```

REMOVE STOPWORDS

In [26]:
```python
# Download stopwords Package to execute this command
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stopwords.words('english')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Owner\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

Out[26]:
```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
 'are',
 'was',
 'were',
 'be',
 'been',
 'being',
 'have',
 'has',
 'had',
 'having',
 'do',
 'does',
 'did',
 'doing',
 'a',
 'an',
 'the',
 'and',
 'but',
 'if',
 'or',
 'because',
 'as',
 'until',
 'while',
 'of',
 'at',
 'by',
 'for',
 'with',
 'about',
 'against',
 'between',
 'into',
 'through',
 'during',
 'before',
 'after',
 'above',
 'below',
 'to',
 'from',
 'up',
 'down',
 'in',
 'out',
 'on',
 'off',
 'over',
 'under',
 'again',
 'further',
 'then',
 'once',
 'here',
 'there',
 'when',
 'where',
 'why',
 'how',
 'all',
 'any',
 'both',
 'each',
 'few',
 'more',
 'most',
 'other',
 'some',
 'such',
 'no',
 'nor',
 'not',
 'only',
 'own',
 'same',
 'so',
 'than',
 'too',
 'very',
 's',
 't',
 'can',
 'will',
 'just',
 'don',
 "don't",
 'should',
 "should've",
 'now',
 'd',
 'll',
 'm',
 'o',
 're',
 've',
 'y',
 'ain',
 'aren',
 "aren't",
 'couldn',
 "couldn't",
 'didn',
 "didn't",
 'doesn',
 "doesn't",
 'hadn',
 "hadn't",
 'hasn',
 "hasn't",
 'haven',
 "haven't",
 'isn',
 "isn't",
 'ma',
 'mightn',
 "mightn't",
 'mustn',
 "mustn't",
 'needn',
 "needn't",
 'shan',
 "shan't",
 'shouldn',
 "shouldn't",
 'wasn',
 "wasn't",
 'weren',
 "weren't",
 'won',
 "won't",
 'wouldn',
 "wouldn't"]
```

In [27]:
```python
Test_punc_removed_join
```

Out[27]:
```
'Hello Mr Future I am so happy to be learning AI now'
```

In [28]:
```python
Test_punc_removed_join_clean = [word for word in Test_punc_removed_join.split() if word.lower() not in stopwords.words('english')]
```

In [29]:
```python
Test_punc_removed_join_clean # Only important (no so common) words are left
```

Out[29]:
```
['Hello', 'Mr', 'Future', 'happy', 'learning', 'AI']
```

In [30]:
```python
# Define a pipeline to clean up all the messages
# The pipeline performs the following: (1) remove punctuation, (2) remove stopwords

def message_cleaning(message):
    Test_punc_removed = [char for char in message if char not in string.punctuation]
    Test_punc_removed_join = ''.join(Test_punc_removed)
    Test_punc_removed_join_clean = [word for word in Test_punc_removed_join.split() if word.lower() not in stopwords.words('english')]
    return Test_punc_removed_join_clean
```

In [31]:
```python
# Test the newly added function
spam_df_clean = spam_df['Text'].apply(message_cleaning)
```

In [32]:
```python
print(spam_df_clean[0])
```

```
['Go', 'jurong', 'point', 'crazy', 'Available', 'bugis', 'n', 'great', 'world', 'la', 'e', 'buffet', 'Cine', 'got', 'amore', 'wat']
```

In [33]:
```python
print(spam_df['Text'][0])
```

```
Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
```

In [34]:
```python
from sklearn.feature_extraction.text import CountVectorizer
# Define the cleaning pipeline we defined earlier
vectorizer = CountVectorizer(analyzer = message_cleaning)
spamham_countvectorizer = vectorizer.fit_transform(spam_df['Text'])
```

In [35]:
```python
print(spamham_countvectorizer.toarray())
```

```
[[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]]
```

In [36]:
```python
spamham_countvectorizer.shape
```

Out[36]:
```
(5572, 11304)
```

TRAINING THE MODEL WITH ALL DATASET

In [37]:
```python
from sklearn.naive_bayes import MultinomialNB

NB_classifier = MultinomialNB()
label = spam_df['Text'].values
NB_classifier.fit(spamham_countvectorizer, label)
```

Out[37]:
```
▾ MultinomialNB
MultinomialNB()
```

In [38]:
```python
testing_sample = ['Free money!!!', "Hi Kim, Please let me know if you need any further information. Thanks"]
testing_sample_countvectorizer = vectorizer.transform(testing_sample)
```

In [39]:
```python
test_predict = NB_classifier.predict(testing_sample_countvectorizer)
test_predict
```

Out[39]:
```
array(["Sorry, I'll call later",
       "Hi! You just spoke to MANEESHA V. We'd like to know if you were satisfied with the experience. Reply Toll Free with Yes or No."],
      dtype='<U910')
```

In [40]:
```python
# Mini Challenge!
testing_sample = ['Hello, I am Ryan, I would like to book a hotel in Bali by January 24th', 'money viagara!!!!!']
```

In [41]:
```python
testing_sample = ['money viagara!!!!!', "Hello, I am Ryan, I would like to book a hotel in SF by January 24th"]
testing_sample_countvectorizer = vectorizer.transform(testing_sample)
test_predict = NB_classifier.predict(testing_sample_countvectorizer)
test_predict
```

Out[41]:
```
array(["Sorry, I'll call later", "Sorry, I'll call later"], dtype='<U910')
```

In [42]:
```python
X = spamham_countvectorizer
y = label
```

In [43]:
```python
X.shape
```

Out[43]:
```
(5572, 11304)
```

In [44]:
```python
y.shape
```

Out[44]:
```
(5572,)
```

In [45]:
```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

In [46]:
```python
from sklearn.naive_bayes import MultinomialNB

NB_classifier = MultinomialNB()
NB_classifier.fit(X_train, y_train)
```

Out[46]:
```
▾ MultinomialNB
MultinomialNB()
```

In [ ]: