

importing the libraries

```
In [1]: import numpy as np
import pandas as pd
import nltk
```

read the data set

```
In [2]: sms=pd.read_csv('spam.csv', encoding='latin-1')
sms.head()
```

	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

```
In [3]: sms=sms.drop(["Unnamed: 2","Unnamed: 3","Unnamed: 4"],axis=1)
sms=sms.rename(columns={"v1":"label","v2":"text"})
sms.head()
```

	label	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...

Explore the data

```
In [4]: print(" no of rows", len(sms))

no of rows 5572
```

```
In [5]: sms.label.value_counts()

ham      4825
spam      747
Name: label, dtype: int64
```

```
In [6]: sms.describe()
```

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

```
In [7]: sms['length']=sms['text'].apply(len)
sms.head()
```

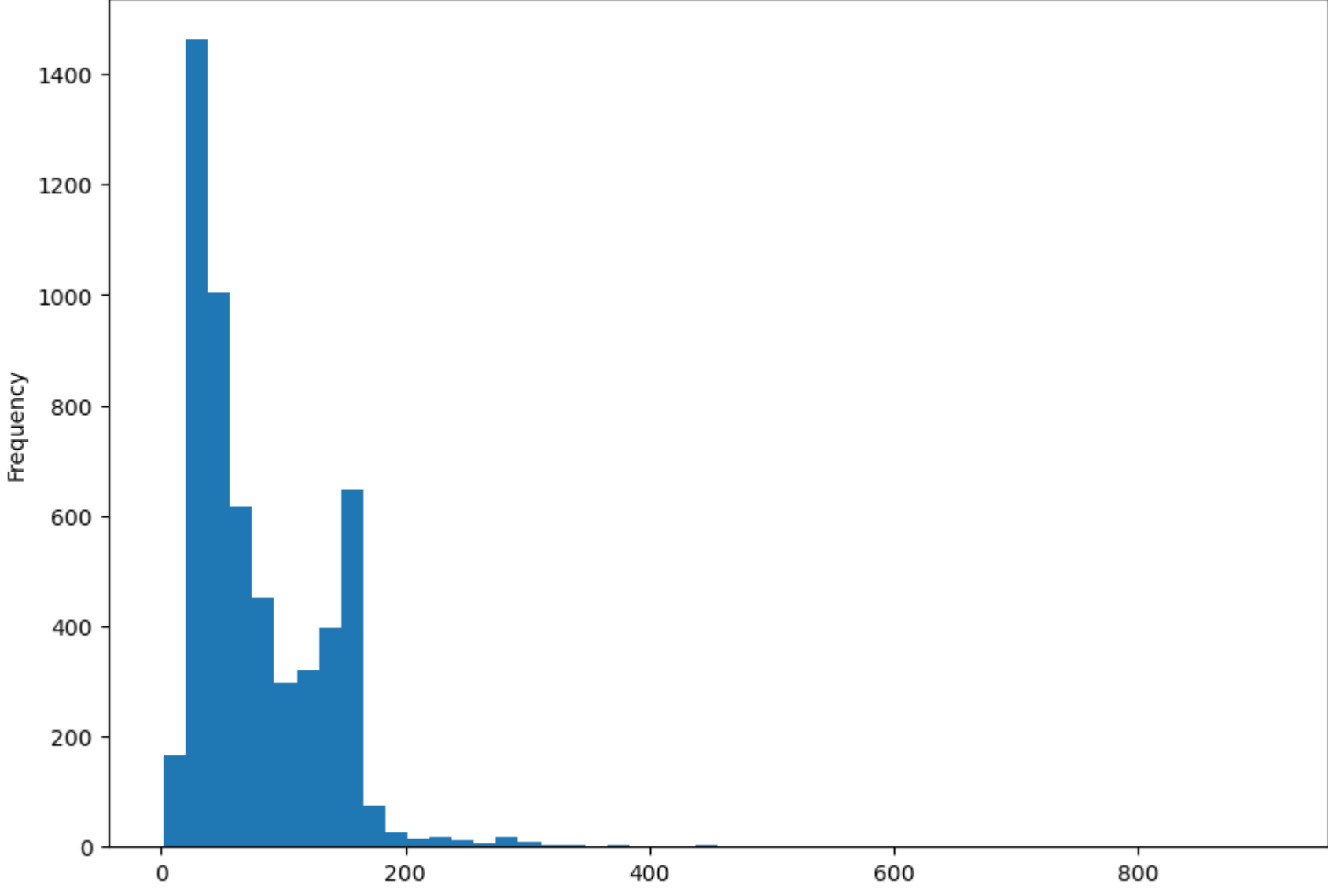
	label	text	length
0	ham	Go until jurong point, crazy.. Available only ...	111
1	ham	Ok lar... Joking wif u oni...	29
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	155
3	ham	U dun say so early hor... U c already then say...	49
4	ham	Nah I don't think he goes to usf, he lives aro...	61

Plot

```
In [8]: import matplotlib.pyplot as plt
import seaborn as sns

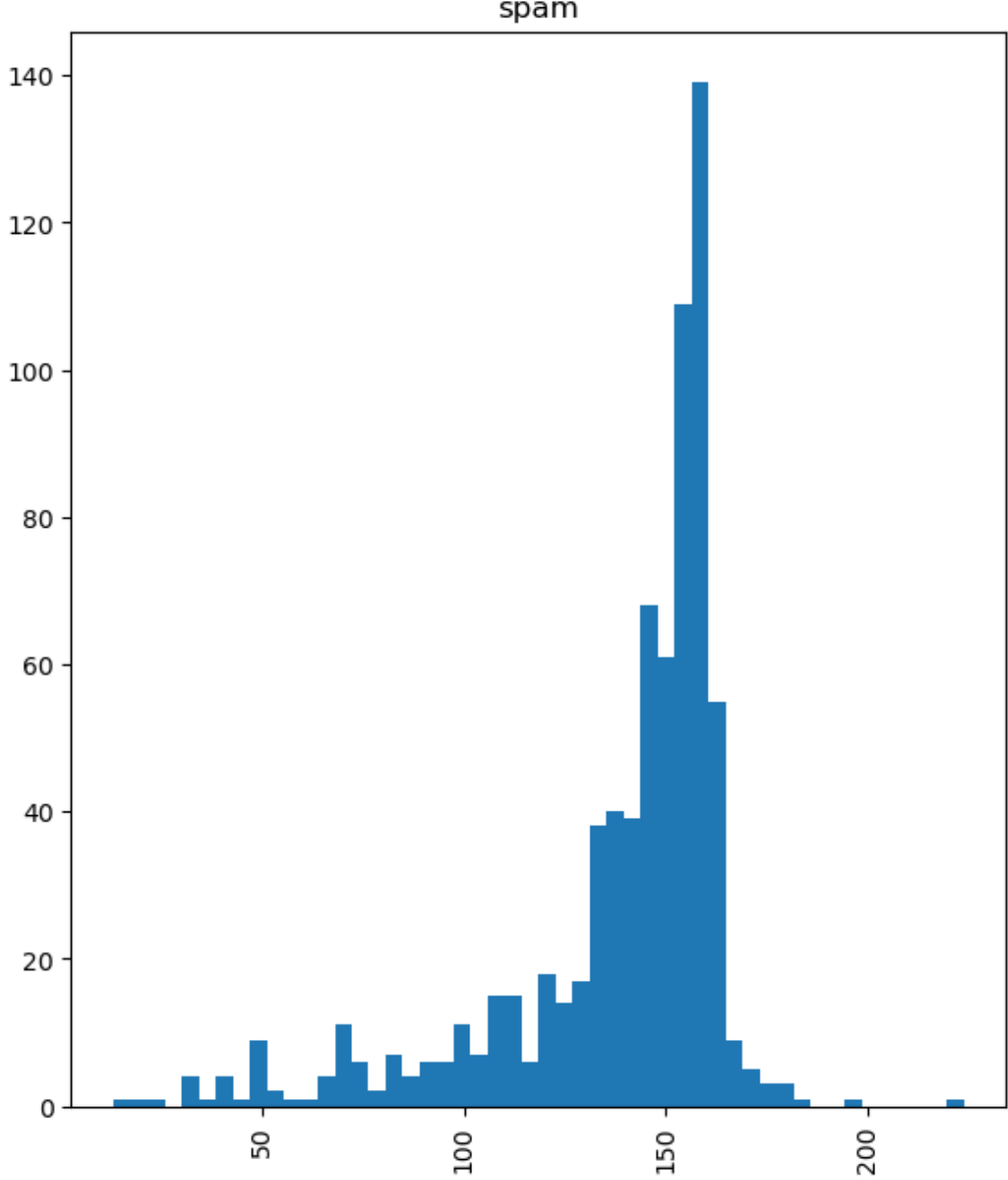
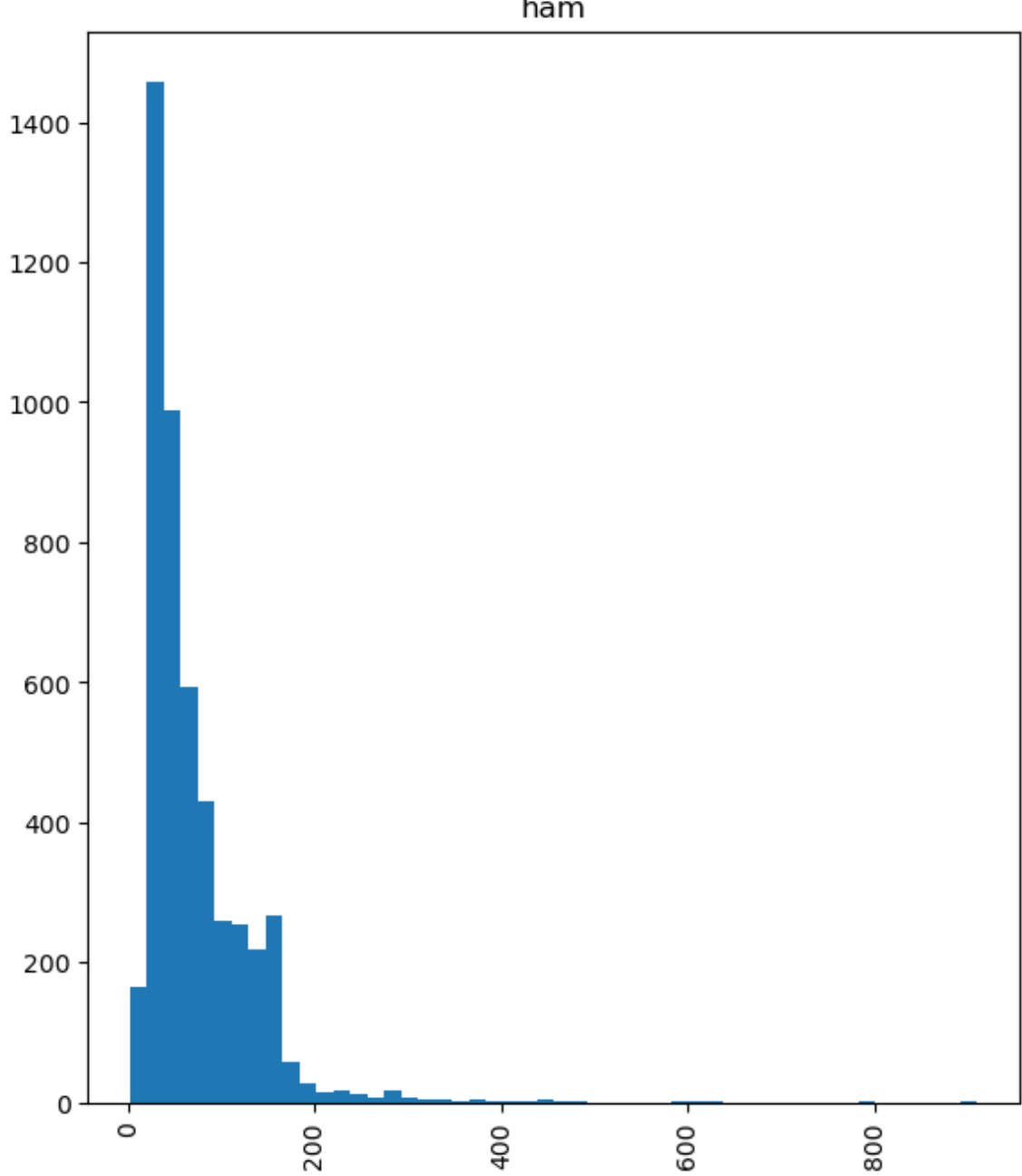
sms['length'].plot(bins=50, kind='hist',figsize=(10,7))
```

<Axes: ylabel='Frequency'>



```
In [9]: sms.hist(column='length', by='label', bins=50, figsize=(15,8))
```

array([<Axes: title='center': 'ham'>, <Axes: title='center': 'spam'>], dtype=object)



```
In [10]: sms.loc[:, 'label']=sms.label.map({'ham':0, 'spam':1})
sms.head()
```

C:\Users\INDIA\AppData\Local\Temp\ipykernel\_7232\1727894437.py:1: DeprecationWarning: In a future version, 'df.iloc[:, i] = newvals' will attempt to set the values inplace instead of always setting a new array. To retain the old behavior, use either 'df[df.columns[i]] = newvals' or, if columns are non-unique, 'df.i

```
setitem(i, newvals)'
  sms.loc[:, 'label']=sms.label.map({'ham':0, 'spam':1})
```

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

Implement Bag of Words

```
In [11]: from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split

count=CountVectorizer()
input=["REMINDER FROM 02: To get 2.50 pounds free call credit and details of great offers pls reply 2 this text with your valid name, house no and postcode"]

text=count.fit_transform(sms['text'])

x_train, x_test, y_train, y_test= train_test_split(text, sms['label'], test_size=0.20, random_state=1)
text
```

<5572x8672 sparse matrix of type '<class 'numpy.int64'>' with 73916 stored elements in Compressed Sparse Row format>

```
In [12]: print(x_train.shape)
print(x_test.shape)

input=text[5571]

(4457, 8672)
(1115, 8672)
```

implementation of ML Model

```
In [13]: from sklearn.neural_network import MLPClassifier

model=MLPClassifier()
model.fit(x_train, y_train)
```

MLPClassifier

```
In [14]: prediction=model.predict(x_test)
print(prediction)

[0 0 0 ... 0 0 0]
```

```
In [15]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

print("Multinomial NB")
print("Accuracy score: {}".format(accuracy_score(y_test, prediction)))
print("Precision score: {}".format(precision_score(y_test, prediction)))
print("Recall score: {}".format(recall_score(y_test, prediction)))
print("F1 score: {}".format(f1_score(y_test, prediction)))

Multinomial NB
Accuracy score: 0.9928251121076234
Precision score: 1.0
Recall score: 0.9424460431654677
F1 score: 0.9703703703703704
```

```
In [16]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

print("Bernoulli NB")
print("Accuracy score: {}".format(accuracy_score(y_test, prediction)))
print("Precision score: {}".format(precision_score(y_test, prediction)))
print("Recall score: {}".format(recall_score(y_test, prediction)))
print("F1 score: {}".format(f1_score(y_test, prediction)))

Bernoulli NB
Accuracy score: 0.9928251121076234
Precision score: 1.0
Recall score: 0.9424460431654677
F1 score: 0.9703703703703704
```

```
In [17]: from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

print("MLP Classifier")
print("Accuracy score: {}".format(accuracy_score(y_test, prediction)))
print("Precision score: {}".format(precision_score(y_test, prediction)))
print("Recall score: {}".format(recall_score(y_test, prediction)))
print("F1 score: {}".format(f1_score(y_test, prediction)))

MLP Classifier
Accuracy score: 0.9928251121076234
Precision score: 1.0
Recall score: 0.9424460431654677
F1 score: 0.9703703703703704
```

input

<1x8672 sparse matrix of type '<class 'numpy.int64'>' with 5 stored elements in Compressed Sparse Row format>

model.predict(input)

array([0], dtype=int64)

```
In [20]: data=sms['text']
input="To get 2.50 pounds free call credit and details of great offers pls reply 2 this text"
```

data

```
Out[21]: 0      Go until jurong point, crazy.. Available only ...
1      Ok lar... Joking wif u oni...
2      Free entry in 2 a wkly comp to win FA Cup fina...
3      U dun say so early hor... U c already then say...
4      Nah I don't think he goes to usf, he lives aro...
...
5567   This is the 2nd time we have tried 2 contact u...
5568   Will I b going to esplanade fr home?
5569   Pity, * was in mood for that. So...any other s...
5570   The guy did some bltching but I acted like i'd...
5571   Rofl. Its true to its name
Name: text, Length: 5572, dtype: object
```

text=count.fit\_transform(data)

text

<5572x8672 sparse matrix of type '<class 'numpy.int64'>' with 73916 stored elements in Compressed Sparse Row format>

input=text[5572]

\*\*\*\*\*End Of The Code\*\*\*\*\*

\*\*\*\*\*Thank You\*\*\*\*\*