In [2]:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
```

In [3]:

```
# data

msg = pd.read_csv('/Users/user/Downloads/spam_not_spam.csv')
msg.shape
```

Out[3]:

```
(5572, 2)
```

In [4]:

```
msg.head()
```

Out[4]:

	Category	Message
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 [5]:

```
features = msg['Message']
target = msg['Category']
```

In [6]:

```
# encode features
# Create a TfidfVectorizer object
vectorizer = TfidfVectorizer()
features_encoded = vectorizer.fit_transform(features)

features_encoded.shape
```

Out[6]:

```
(5572, 8709)
```

In [7]:

```
# encode target as well for binary crossentropy
# Create a LabelEncoder object
label_encoder = LabelEncoder()
target_encoded = label_encoder.fit_transform(target)

target_encoded.shape
```

Out[7]:

```
(5572,)
```

In [8]:

```
# create a train and a test split
X_train, X_test, y_train, y_test = train_test_split(features_encoded, target_encoded, test_size=0.2, random_s
```

In [9]:

```
# using nerual networks

mlp = MLPClassifier(hidden_layer_sizes=(100,100,10), max_iter=100, random_state=42)
mlp.fit(X_train, y_train)
```

Out[9]:

```
MLPClassifier(hidden_layer_sizes=(100, 100, 10), max_iter=100, random_state=42)
```

In [11]:

```
mlp.score(X_test,y_test)  # test score 99% accuracy
```

Out[11]:

```
0.9910313901345291
```

In [12]:

```
mlp.score(X_train,y_train) # train score 100% accuracy
```

Out[12]:

```
1.0
```

In [51]:

```
#classification report score

from sklearn.metrics import classification_report

y_pred = mlp.predict(X_test)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99       966
           1       1.00      0.93      0.97       149

    accuracy                           0.99      1115
   macro avg       0.99      0.97      0.98      1115
weighted avg       0.99      0.99      0.99      1115
```

In [ ]:

```

```