In [1]:

```
#import required library
import numpy as np
import pandas as pd
```

In [4]:

```
import pandas as pd
dataset = pd.read_csv("spam.csv")
dataset.read_csv(spam.csv, encoding="utf-8")
dataset.head()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-4-a3630bb4fc20> in <module>
      1 import pandas as pd
----> 2 dataset = pd.read_csv("spam.csv")
      3 dataset.read_csv(spam.csv, encoding="utf-8")
      4 dataset.head()

/usr/local/lib/python3.7/dist-packages/pandas/util/_decorators.py in wrapper(*args, **kwa
rgs)
    309                     stacklevel=stacklevel,
    310                 )
--> 311             return func(*args, **kwargs)
    312
    313         return wrapper

/usr/local/lib/python3.7/dist-packages/pandas/io/parsers/readers.py in read_csv(filepath_
or_buffer, sep, delimiter, header, names, index_col, usecols, squeeze, prefix, mangle_dup
e_cols, dtype, engine, converters, true_values, false_values, skipinitialspace, skiprows,
skipfooter, nrows, na_values, keep_default_na, na_filter, verbose, skip_blank_lines, pars
e_dates, infer_datetime_format, keep_date_col, date_parser, dayfirst, cache_dates, iterat
or, chunksize, compression, thousands, decimal, lineterminator, quotechar, quoting, doubl
equote, escapechar, comment, encoding, encoding_errors, dialect, error_bad_lines, warn_ba
d_lines, on_bad_lines, delim_whitespace, low_memory, memory_map, float_precision, storage
_options)
    584     kwds.update(kwds_defaults)
    585
--> 586     return _read(filepath_or_buffer, kwds)
    587
    588

/usr/local/lib/python3.7/dist-packages/pandas/io/parsers/readers.py in _read(filepath_or_
buffer, kwds)
    480
    481     # Create the parser.
--> 482     parser = TextFileReader(filepath_or_buffer, **kwds)
    483
    484     if chunksize or iterator:

/usr/local/lib/python3.7/dist-packages/pandas/io/parsers/readers.py in __init__(self, f,
engine, **kwds)
    809             self.options["has_index_names"] = kwds["has_index_names"]
    810
--> 811         self._engine = self._make_engine(self.engine)
    812
    813     def close(self):

/usr/local/lib/python3.7/dist-packages/pandas/io/parsers/readers.py in _make_engine(self,
engine)
   1038             )
   1039         # error: Too many arguments for "ParserBase"
-> 1040         return mapping[engine](self.f, **self.options)  # type: ignore[call-arg]
   1041
   1042     def _failover_to_python(self):

/usr/local/lib/python3.7/dist-packages/pandas/io/parsers/c_parser_wrapper.py in __init__(
```

```
self, src, **kwds)
     49
     50         # open handles
---> 51         self._open_handles(src, kwds)
     52         assert self.handles is not None
     53

/usr/local/lib/python3.7/dist-packages/pandas/io/parsers/base_parser.py in _open_handles(
self, src, kwds)
    227             memory_map=kwds.get("memory_map", False),
    228             storage_options=kwds.get("storage_options", None),
--> 229             errors=kwds.get("encoding_errors", "strict"),
    230         )
    231

/usr/local/lib/python3.7/dist-packages/pandas/io/common.py in get_handle(path_or_buf, mod
e, encoding, compression, memory_map, is_text, errors, storage_options)
    705                 encoding=ioargs.encoding,
    706                 errors=errors,
--> 707                 newline="",
    708             )
    709         else:

FileNotFoundError: [Errno 2] No such file or directory: 'spam.csv'
```

In [5]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
```

In [6]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
```

In [7]:

```
model = Sequential()
model.add(LSTM(50, input_shape=(60, 1),return_sequences=True))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50,return_sequences=True))
model.add(Dense(1))
model.add(Dense(300,activation='relu')) # Hidden layer 1
model.add(Dense(150,activation='relu')) # Hidden layer 2
model.add(Dense(4,activation='softmax')) # Output layer
```

In [8]:

```
#Compile the model
model.compile(optimizer='adam',loss='mse')
```

In [11]:

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random_state=4
2)
model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=10,
                    validation_data=xtest,
                    validation_steps=len(xtest))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-b42eb16b5ba3> in <module>
      1 from sklearn.model_selection import train_test_split
----> 2 x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.33, random
_state=42)
      3 model.fit_generator(xtrain,
```

```
4                     steps_per_epoch=len(xtrain),
5                     epochs=10,
```

```
NameError: name 'x' is not defined
```

In [10]:

```
#Save the model
model.save('LSTM.h5')
```