In [1]:

```
import numpy as np
import pandas as pd
import sklearn.model_selection as model_selection
import sklearn.preprocessing as preproc
from keras.utils import to_categorical
```

```
Using TensorFlow backend.
```

In [82]:

```
dat = pd.read_csv("tic_date_emb_label.csv")
```

In [83]:

```
dat.head()
```

Out[83]:

	Unnamed: 0	0	1	2	3	4	5	6	7	8
0	0	GOOGL	2016-08-22	-1.733884	-0.921825	1.429187	1.680829	-1.369395	1.166707	0.725
1	1	MSFT	2014-07-29	-0.356478	-0.576423	-0.474687	0.661476	0.429892	0.293570	-0.883
2	2	IBM	2013-11-14	0.827591	-0.461297	0.495272	-0.159535	-0.971966	-1.067975	-1.011
3	3	MSFT	2015-08-20	0.033768	0.028286	-1.223439	-0.869134	0.059433	0.552658	-2.039
4	4	MSFT	2016-12-20	-1.437074	-1.608165	-0.581954	-1.303739	1.081447	0.349383	0.474

5 rows × 104 columns

In [211]:

```
x = np.array(dat.iloc[:, 3:103])
y = np.array(dat.iloc[:, 103])
```

In [212]:

```
# scale and split
x = preproc.scale(x, axis = 1)
x = x.reshape(x.shape[0], 10, 10)
y = to_categorical(y)
x_train, x_test, y_train, y_test = model_selection.train_test_split(x, y,
test_size = 0.25)
```

In [221]:

```
import keras
```

```
from keras.models import Sequential
from keras.layers import Dense, Activation, LSTM, Dropout
from keras.layers.convolutional import Conv2D
```

In [222]:

```
# create the model
model = Sequential()
model.add(LSTM(48, input_shape = x_train.shape[1:]))
model.add(Dropout(0.8))
model.add(Dense(2, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['
accuracy'])
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_57 (LSTM)               (None, 48)                11328
_________________________________________________________________
dropout_56 (Dropout)         (None, 48)                0
_________________________________________________________________
dense_54 (Dense)             (None, 2)                 98
=================================================================
Total params: 11,426
Trainable params: 11,426
Non-trainable params: 0
_________________________________________________________________
```

In [223]:

```
# Fit the model
history = model.fit(x_train, y_train,
          validation_data = (x_test, y_test),
          epochs = 40,
          batch_size = 64)
```

```
Train on 2087 samples, validate on 696 samples
Epoch 1/40
2087/2087 [==============================] - 4s - loss: 0.7033 - acc: 0.522
8 - val_loss: 0.6981 - val_acc:
0.4828
```

```
Epoch 2/40
2087/2087 [==============================] - 0s - loss: 0.7070 - acc: 0.499
8 - val_loss: 0.6965 - val_acc:
0.4943
```

```
Epoch 3/40
2087/2087 [==============================] - 0s - loss: 0.7045 - acc: 0.503
6 - val_loss: 0.6959 - val_acc:
0.5115

Epoch 4/40
2087/2087 [==============================] - 0s - loss: 0.6981 - acc: 0.514
6 - val_loss: 0.6962 - val_acc:
0.5014

Epoch 5/40
2087/2087 [==============================] - 0s - loss: 0.6948 - acc: 0.527
1 - val_loss: 0.6958 - val_acc:
0.5144
```

```
Epoch 6/40
2087/2087 [==============================] - 1s - loss: 0.6984 - acc: 0.516
1 - val_loss: 0.6947 - val_acc:
0.5187

Epoch 7/40
2087/2087 [==============================] - 0s - loss: 0.6964 - acc: 0.515
1 - val_loss: 0.6949 - val_acc:
0.5244

Epoch 8/40
2087/2087 [==============================] - 0s - loss: 0.6967 - acc: 0.514
1 - val_loss: 0.6937 - val_acc:
0.5216

Epoch 9/40
2087/2087 [==============================] - 0s - loss: 0.6940 - acc: 0.518
4 - val_loss: 0.6947 - val_acc:
0.5101
```

```
Epoch 10/40
2087/2087 [==============================] - 0s - loss: 0.6931 - acc: 0.526
1 - val_loss: 0.6936 - val_acc:
0.5244
```

```
Epoch 11/40
2087/2087 [==============================] - 1s - loss: 0.6901 - acc: 0.538
6 - val_loss: 0.6937 - val_acc:
0.5144
```

```
Epoch 12/40
2087/2087 [==============================] - 0s - loss: 0.6933 - acc: 0.526
1 - val_loss: 0.6945 - val_acc:
0.5144
```

```
Epoch 13/40
2087/2087 [==============================] - 0s - loss: 0.6922 - acc: 0.519
9 - val_loss: 0.6949 - val_acc:
0.5101

Epoch 14/40
2087/2087 [==============================] - 0s - loss: 0.6888 - acc: 0.564
9 - val_loss: 0.6947 - val_acc:
0.5043

Epoch 15/40
2087/2087 [==============================] - 1s - loss: 0.6906 - acc: 0.534
3 - val_loss: 0.6943 - val_acc:
0.5101
```

```
Epoch 16/40
2087/2087 [==============================] - 1s - loss: 0.6861 - acc: 0.533
8 - val_loss: 0.6944 - val_acc:
0.5101

Epoch 17/40
2087/2087 [==============================] - 1s - loss: 0.6894 - acc: 0.541
4 - val_loss: 0.6955 - val_acc:
0.5057

Epoch 18/40
2087/2087 [==============================] - 1s - loss: 0.6896 - acc: 0.528
0 - val_loss: 0.6958 - val_acc:
0.5043

Epoch 19/40
2087/2087 [==============================] - 0s - loss: 0.6903 - acc: 0.535
7 - val_loss: 0.6953 - val_acc:
0.5072
```

```
Epoch 20/40
2087/2087 [==============================] - 0s - loss: 0.6879 - acc: 0.533
8 - val_loss: 0.6961 - val_acc:
0.5072
```

```
Epoch 21/40
2087/2087 [==============================] - 0s - loss: 0.6893 - acc: 0.541
4 - val_loss: 0.6961 - val_acc:
0.5000
```

```
Epoch 22/40
2087/2087 [==============================] - 0s - loss: 0.6862 - acc: 0.534
7 - val_loss: 0.6949 - val_acc:
0.5129
```

```
Epoch 23/40
2087/2087 [==============================] - 0s - loss: 0.6846 - acc: 0.549
1 - val_loss: 0.6973 - val_acc:
0.5043

Epoch 24/40
2087/2087 [==============================] - 0s - loss: 0.6857 - acc: 0.555
8 - val_loss: 0.6963 - val_acc:
0.5086

Epoch 25/40
2087/2087 [==============================] - 1s - loss: 0.6808 - acc: 0.562
5 - val_loss: 0.6961 - val_acc:
0.5029

Epoch 26/40
```

Epoch 26/40
2087/2087 [==============================] - 0s - loss: 0.6807 - acc: 0.554
4 - val_loss: 0.6970 - val_acc:
0.5101

Epoch 27/40
2087/2087 [==============================] - 0s - loss: 0.6795 - acc: 0.558
2 - val_loss: 0.6975 - val_acc:
0.5014

Epoch 28/40
2087/2087 [==============================] - 0s - loss: 0.6798 - acc: 0.569
2 - val_loss: 0.6966 - val_acc:
0.5187

Epoch 29/40
2087/2087 [==============================] - 1s - loss: 0.6760 - acc: 0.560
1 - val_loss: 0.7041 - val_acc:
0.4885

```
Epoch 30/40
2087/2087 [==============================] - 0s - loss: 0.6710 - acc: 0.585
5 - val_loss: 0.7067 - val_acc:
0.5057

Epoch 31/40
2087/2087 [==============================] - 0s - loss: 0.6687 - acc: 0.585
1 - val_loss: 0.6986 - val_acc:
0.4971

Epoch 32/40
2087/2087 [==============================] - 1s - loss: 0.6669 - acc: 0.584
1 - val_loss: 0.7053 - val_acc:
0.4957
```

```
Epoch 33/40
2087/2087 [==============================] - 1s - loss: 0.6632 - acc: 0.591
8 - val_loss: 0.7057 - val_acc:
0.5316

Epoch 34/40
2087/2087 [==============================] - 1s - loss: 0.6647 - acc: 0.595
1 - val_loss: 0.7116 - val_acc:
0.5216

Epoch 35/40
2087/2087 [==============================] - 0s - loss: 0.6549 - acc: 0.608
5 - val_loss: 0.7112 - val_acc:
0.5244

Epoch 36/40
```

2087/2087 [==============================] - 0s - loss: 0.6566 - acc: 0.598
9 - val_loss: 0.7130 - val_acc:
0.5216

Epoch 37/40
2087/2087 [==============================] - 0s - loss: 0.6461 - acc: 0.623
9 - val_loss: 0.7239 - val_acc:
0.5086

Epoch 38/40
2087/2087 [==============================] - 0s - loss: 0.6472 - acc: 0.633
0 - val_loss: 0.7234 - val_acc:
0.5230

Epoch 39/40
2087/2087 [==============================] - 1s - loss: 0.6397 - acc: 0.641
6 - val_loss: 0.7348 - val_acc:
0.5029

```
Epoch 40/40
2087/2087 [==============================] - 0s - loss: 0.6417 - acc: 0.621
0 - val_loss: 0.7371 - val_acc:
0.4928
```

In [224]:

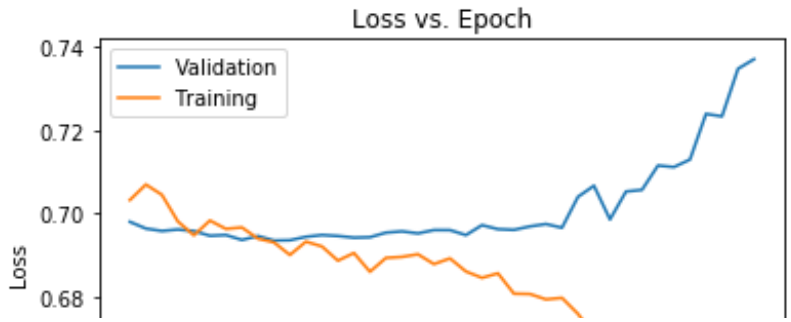
```
val_loss = history.history['val_loss']
loss = history.history['loss']
acc = history.history['acc']
val_acc = history.history['val_acc']
```

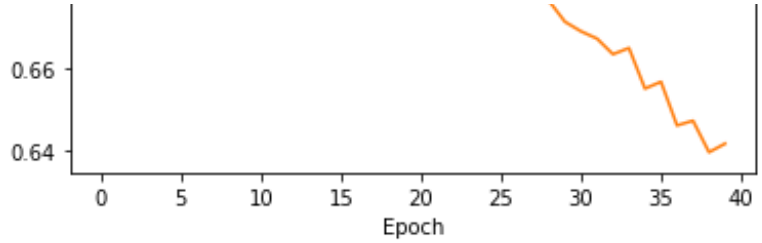
In [225]:

```
import matplotlib.pyplot as plt
```

In [239]:

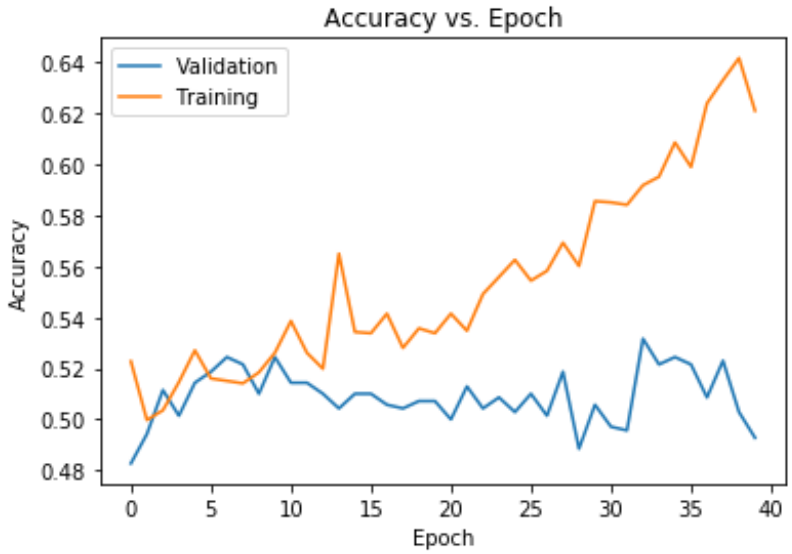
```
plt.plot(val_loss, label = "Validation")
plt.plot(loss, label = "Training")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.title("Loss vs. Epoch")
plt.legend()
plt.show()
```





In [238]:

```
plt.plot(val_acc, label = "Validation")
plt.plot(acc, label = "Training")
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.title("Accuracy vs. Epoch")
plt.legend()
plt.show()
```



In [ ]:

```

```