## **Team 7**

Lei Liu, Yuchen He, Xiangyu Chen

### **Models we are using**

1. BOW model

2. Word embeddings (GLOVE)

3. RNN

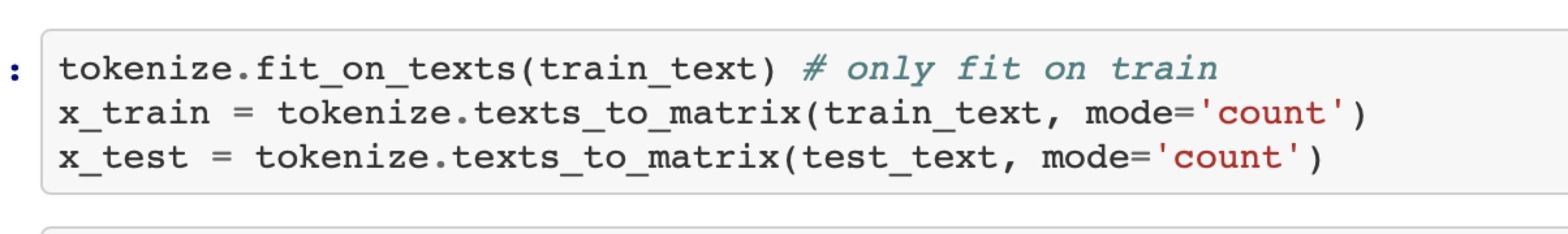
## **Experiment 1**

**1. Bag of word**

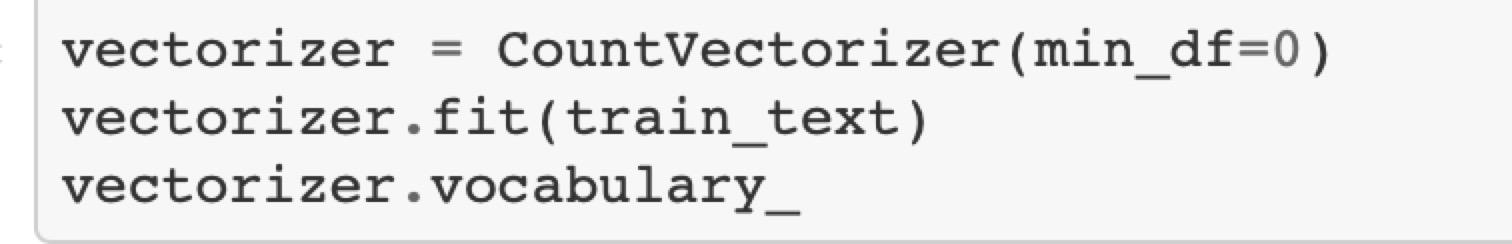
Bag of words model: assumption for a text, ignore the word order and grammar, syntax, it just as a word set, or the term of a combination of, the emergence of each word in the text are independent, does not depend on whether the other word, or when the author of this article choose a word in an arbitrary position is not affected by the previous sentence and independent choice.

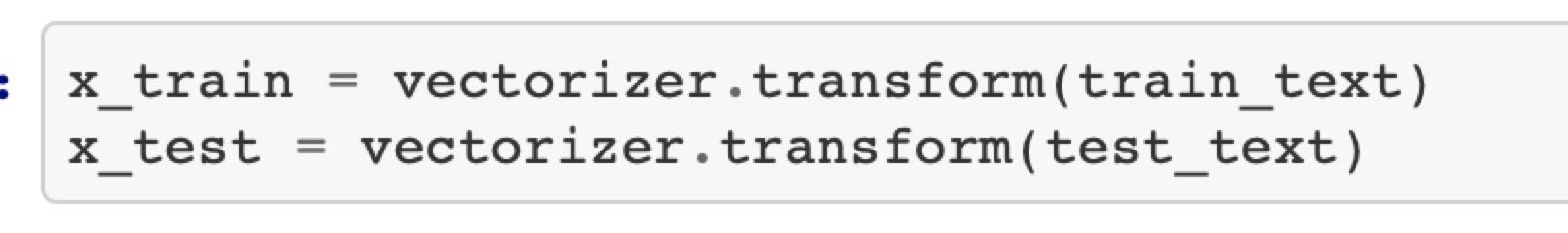
In our model, we experiment on two methods to encode our dataset.

1.

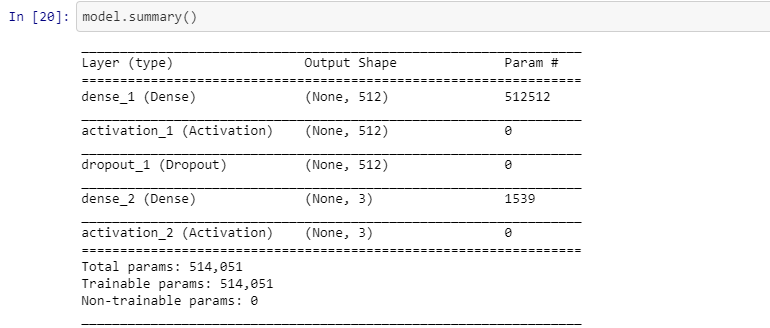


2.

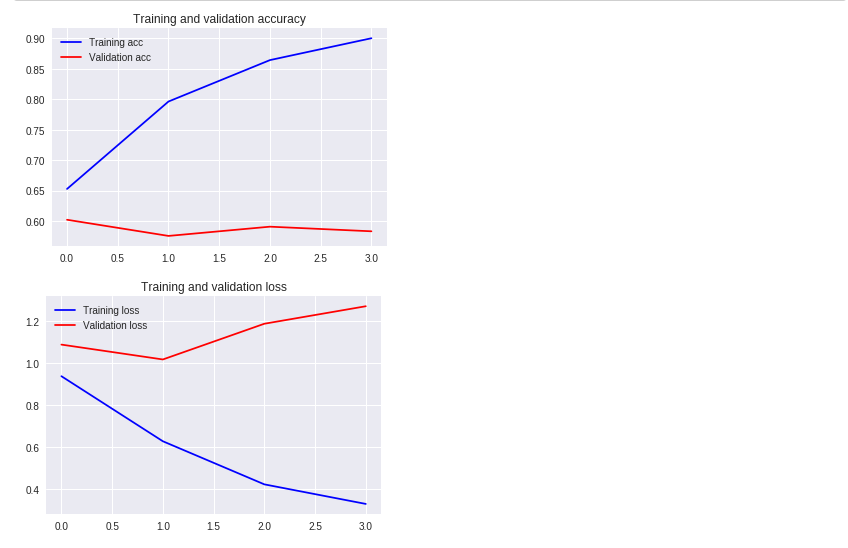




**Model summary:**



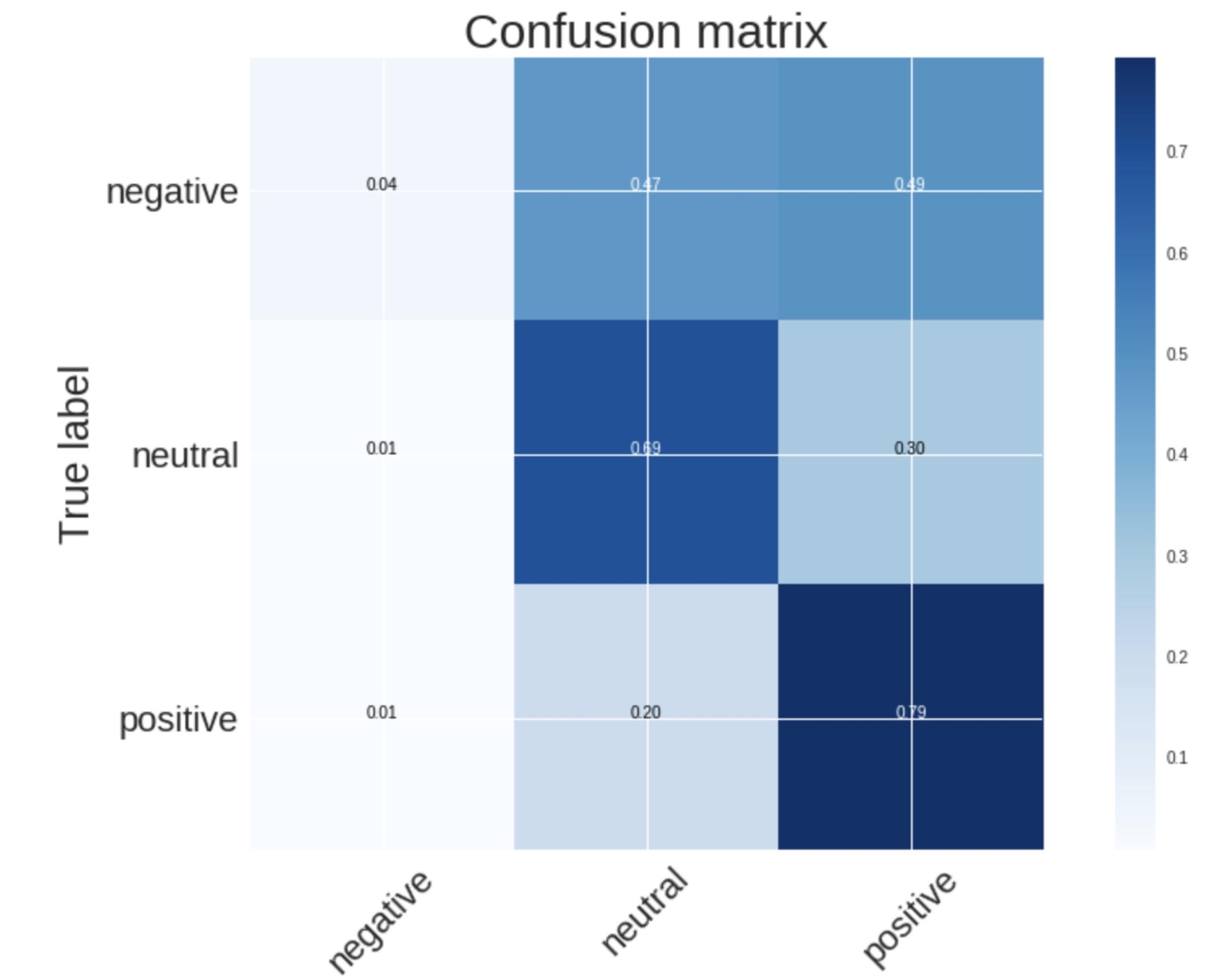
**Visualization:**



**Result:**

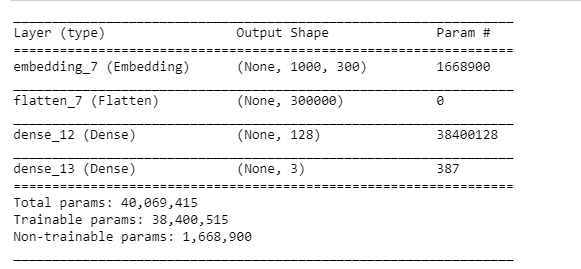


**Confusion Matrix:**

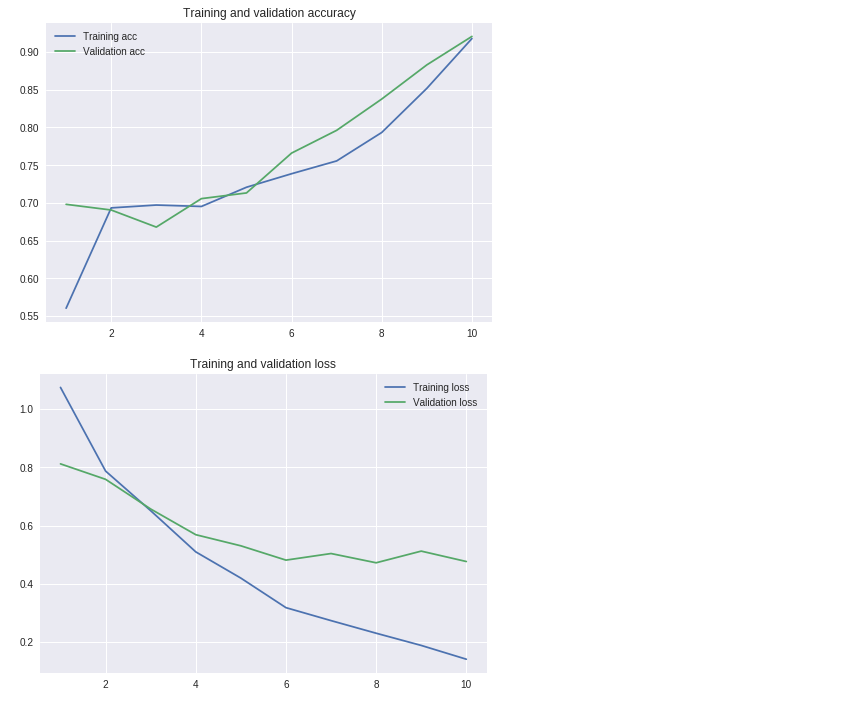
****

**2. Word embeddings**

Model summary:



Visualization:



Result:

Test loss = 0.4767

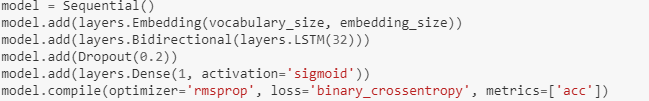
Test accuracy = 0.5833

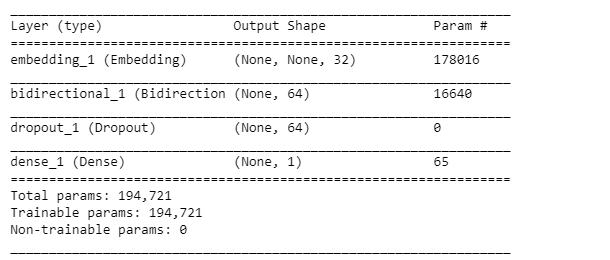
**3. RNN**

The idea behind rnn is make use of sequential information. In other words, if we want to predict a word, we want to know which words came before it. And we calculate every sequence through all tasks. What we use is LSTM which is a extension of RNN model itself. LSTM helps rnn remembers its inputs for quite amount of time. It is like a gate cell that decide whether or not to store or delete the data. Bidirectional rnn is a rnn that contains two hidden layers of opposite directions to the same output, in order to get more information to the network.

For modeling, we use bidirectional rnn with Long short-term memory to perform sentiment analysis.

Model summary:

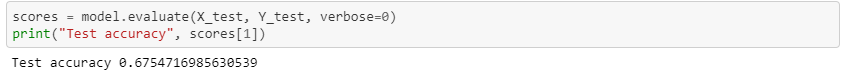




Visualization:



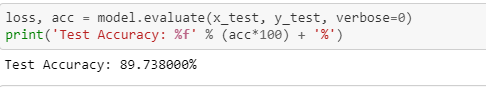
Result:



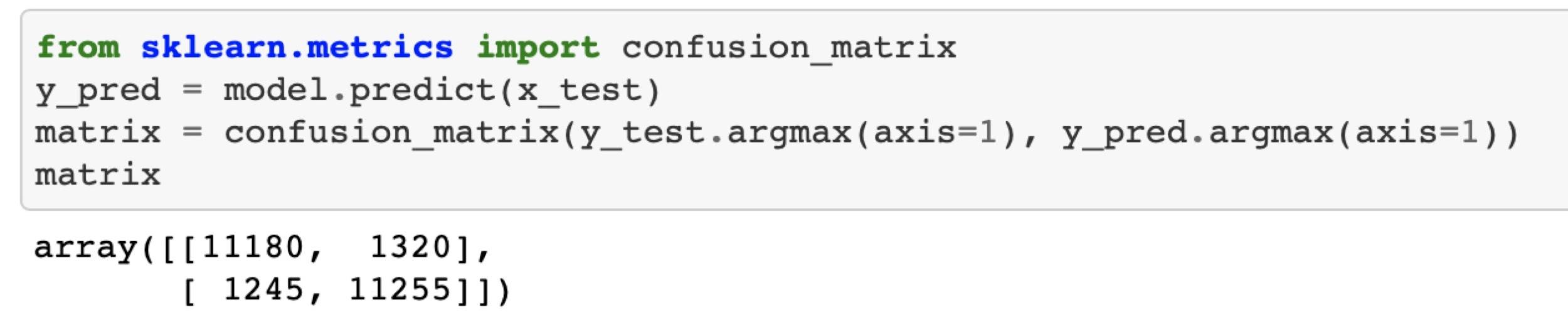
## **Experiment 2: Transfer learning**

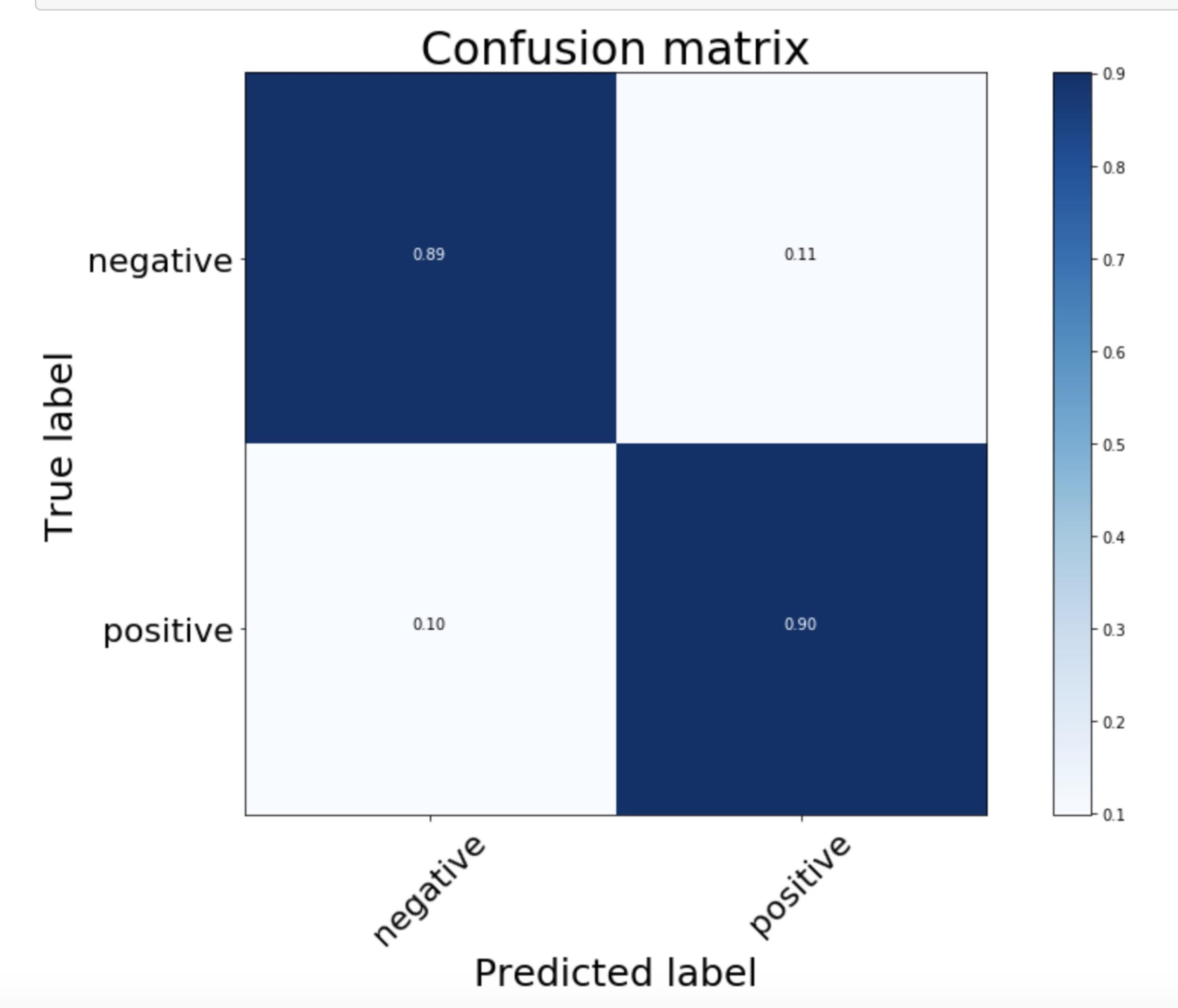
**1. Bag of Word model**

Test accuracy:



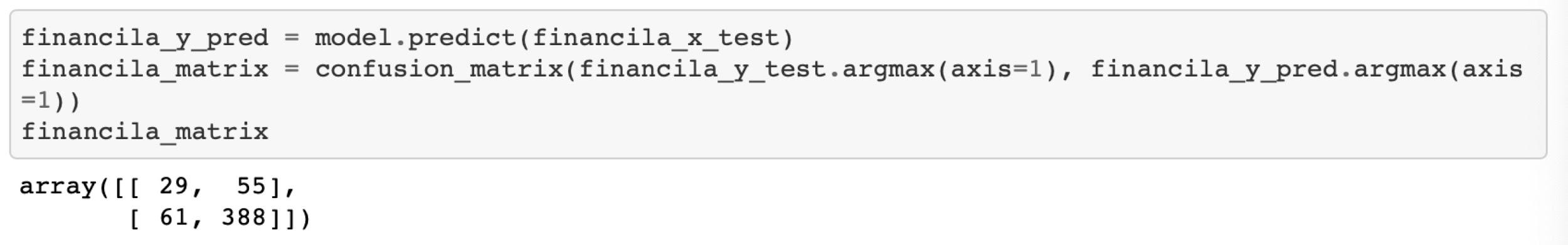
Confusion Matrix:

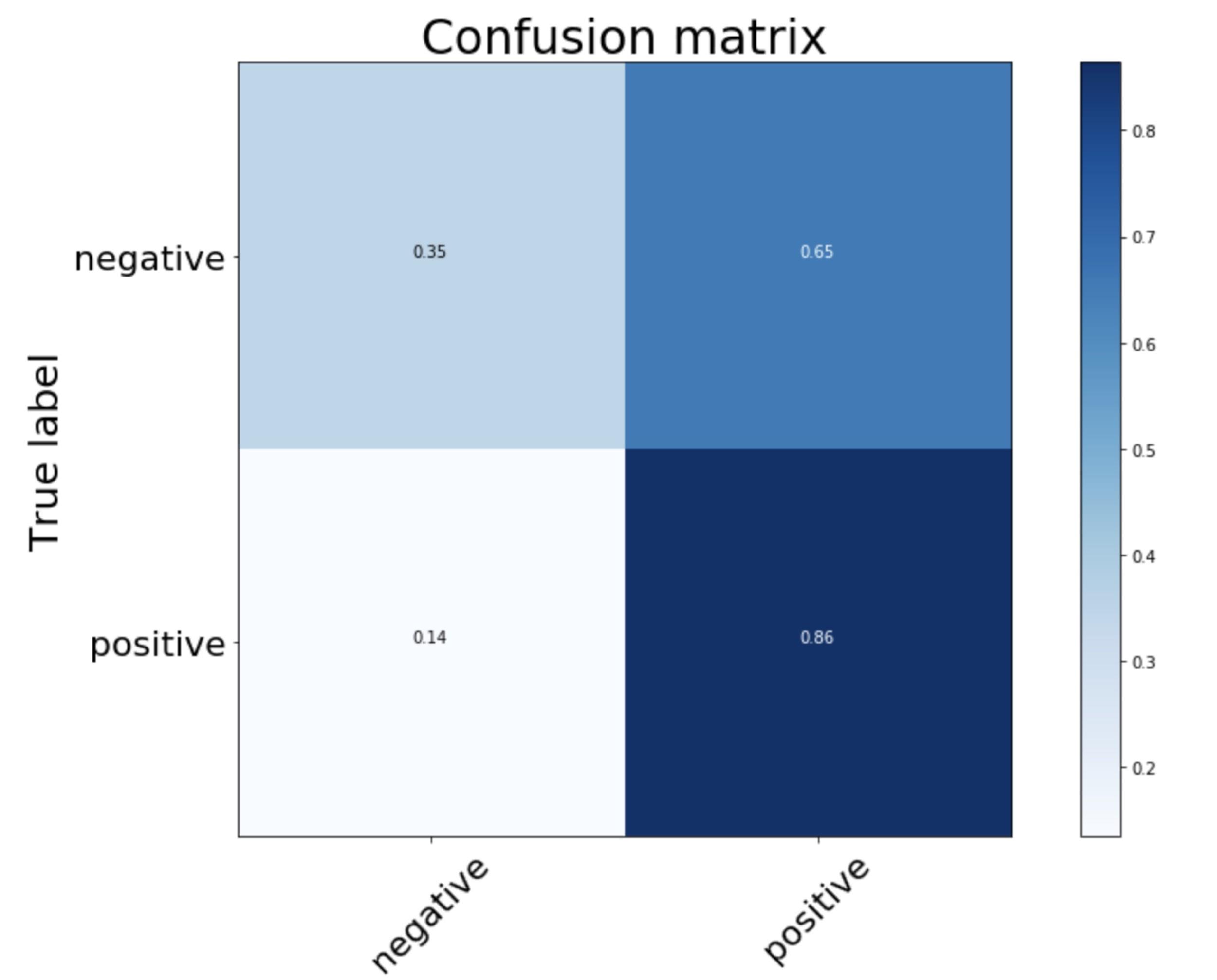




Test on financial dataset(remove neutral labels):

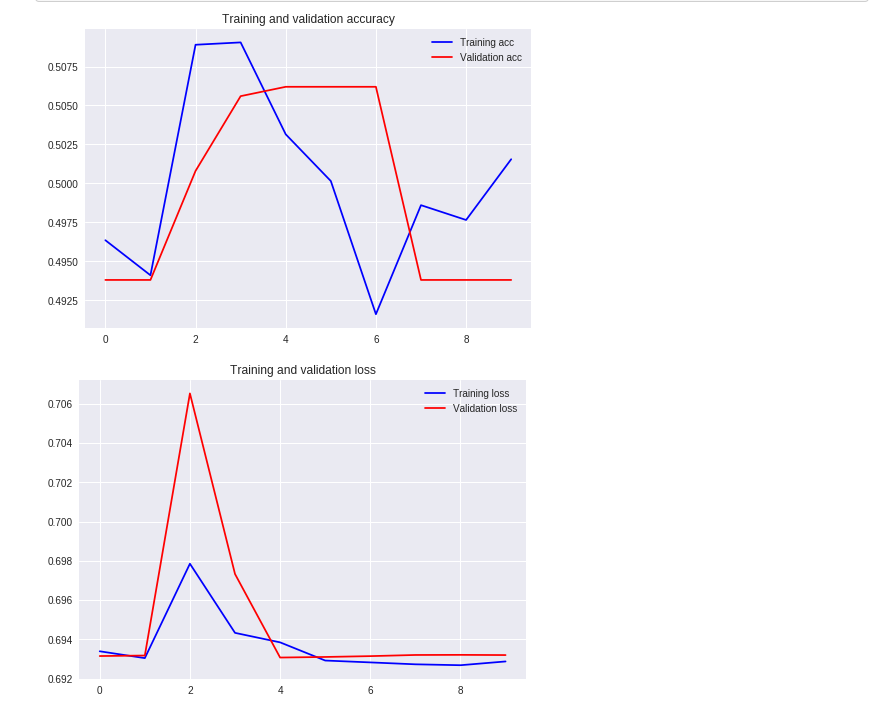






**2. Word Embedding**

Visualization:

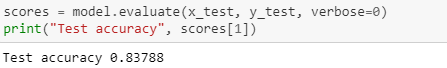


Result:

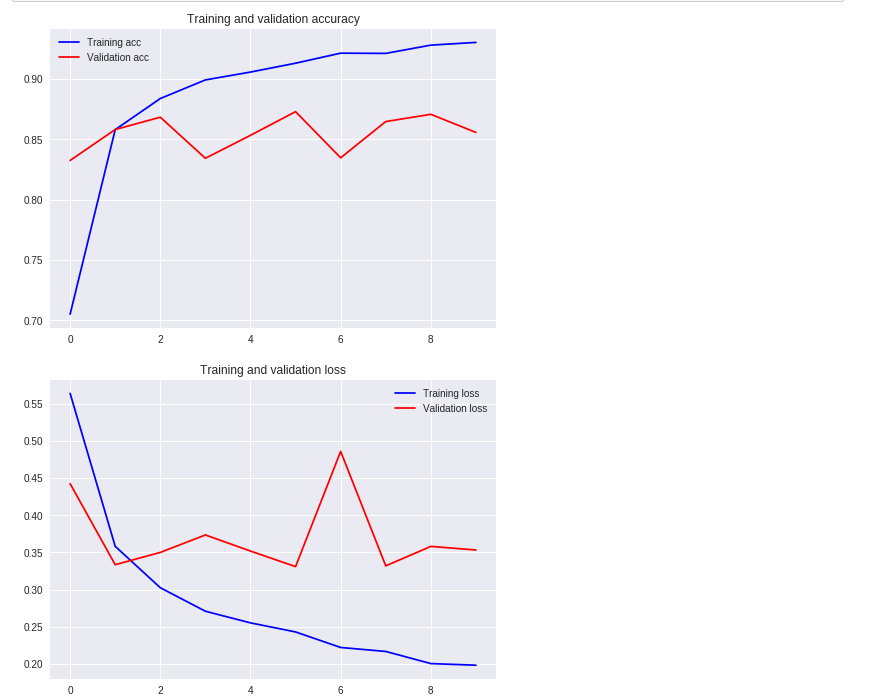


**3. RNN**

Test accuracy:

****

Visualization:

****

## 

## 

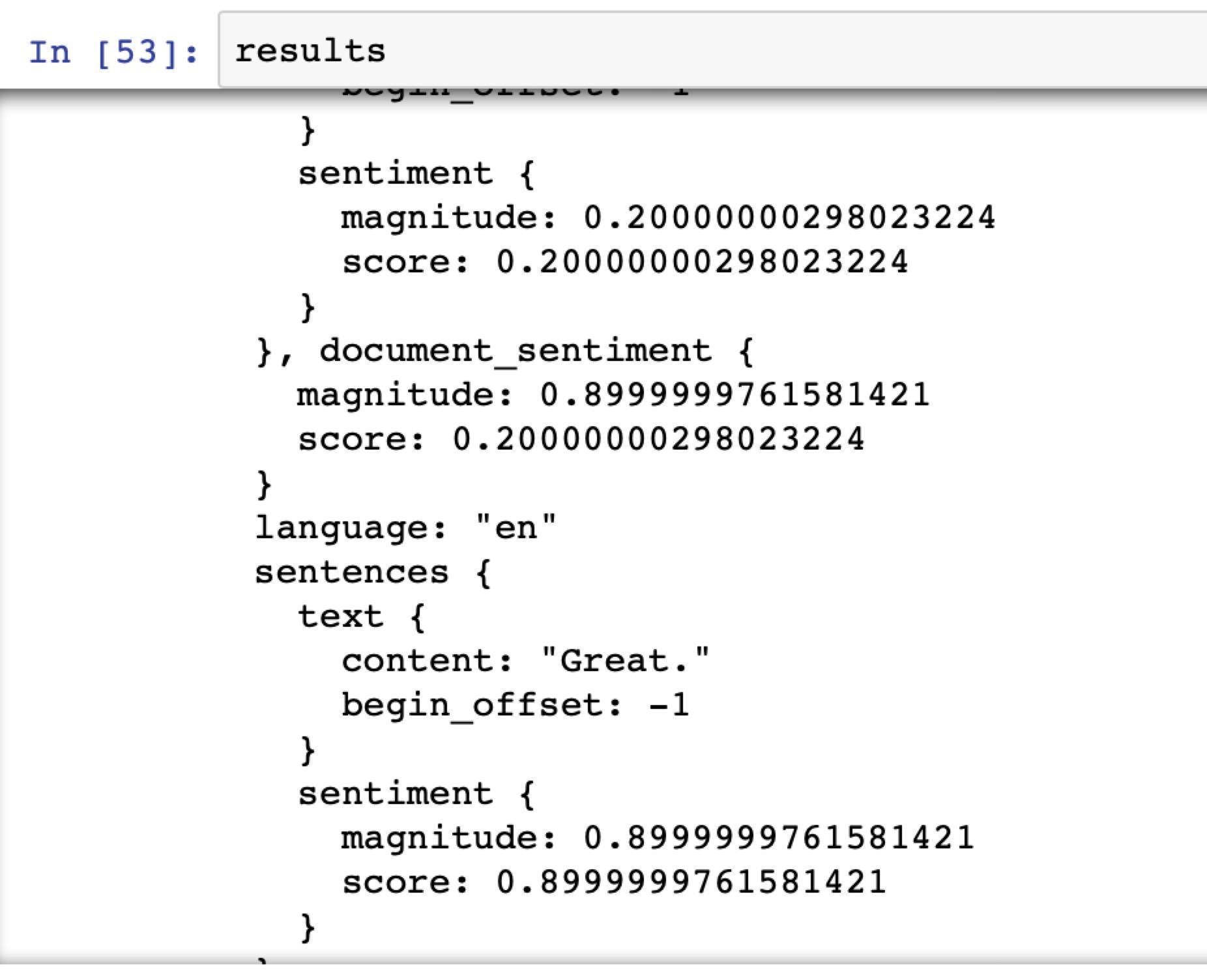
## 

## 

## 

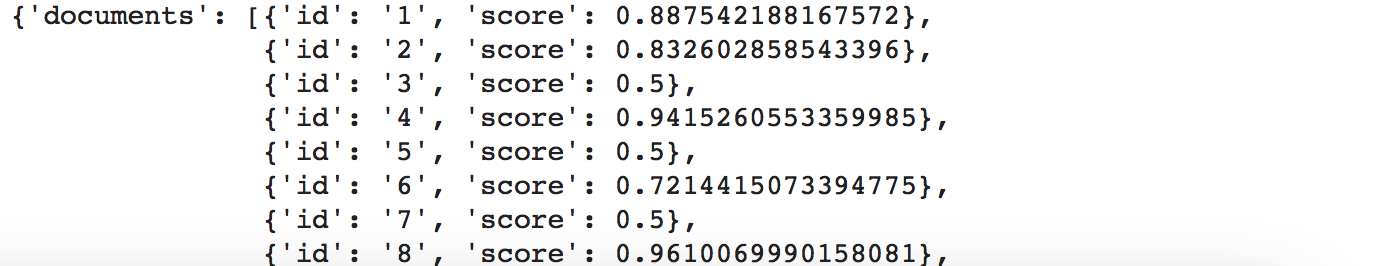
## **Experiment 3: Using APIs**

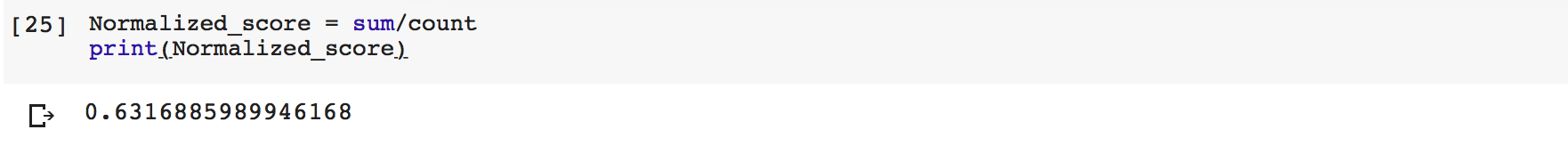
**1. Google api**

****

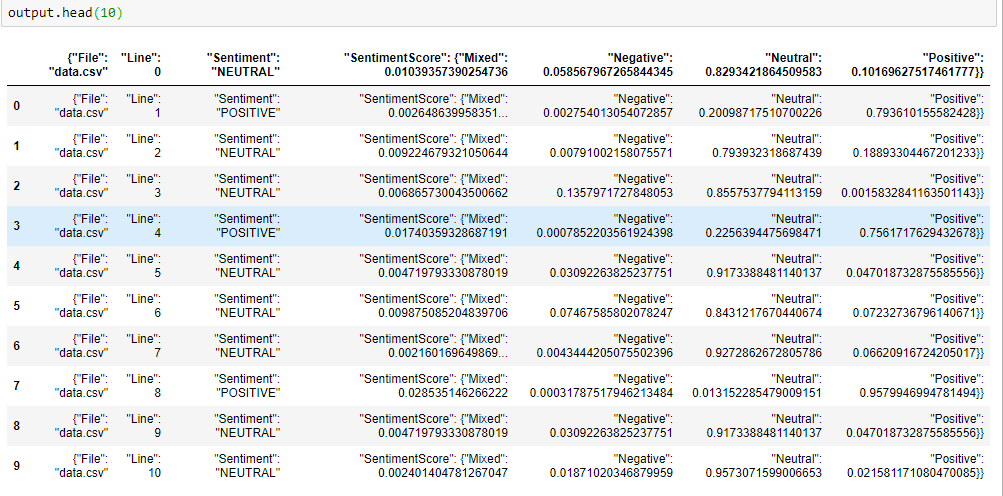
****

**2. Microsoft Azure api**

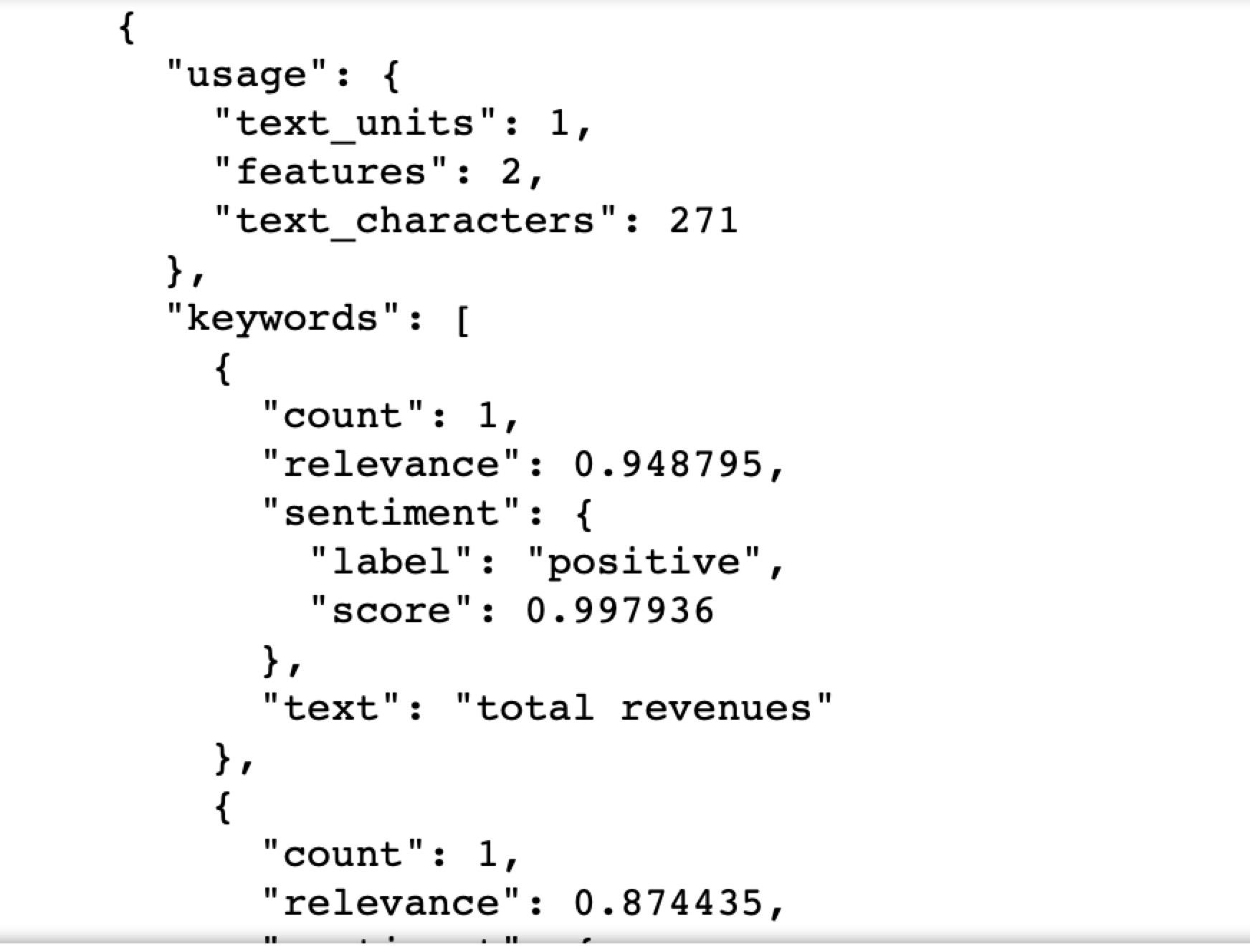
****

****

**3. Amazon api.**

****

**4. Watson api**

****

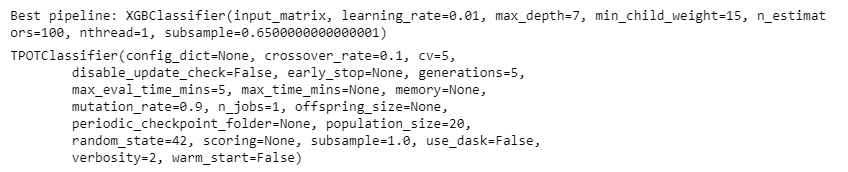
****

**Average normalized score for the database s 0.31600551006.**

**Experiment 4: Ensemble learning using AutoML**

**1. TPOT**

Best model by far, due to early stop



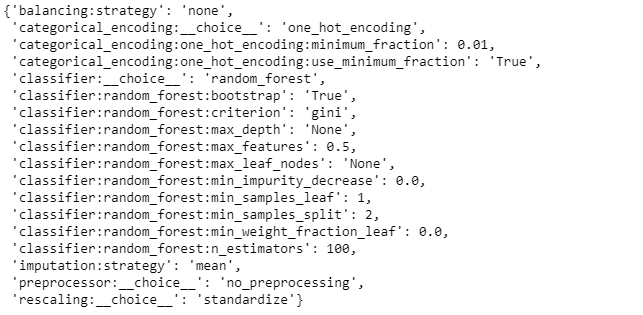
Result:



**This model is better than metrics from Experiment 3**

**2. Auto SKLearn**

Best model:

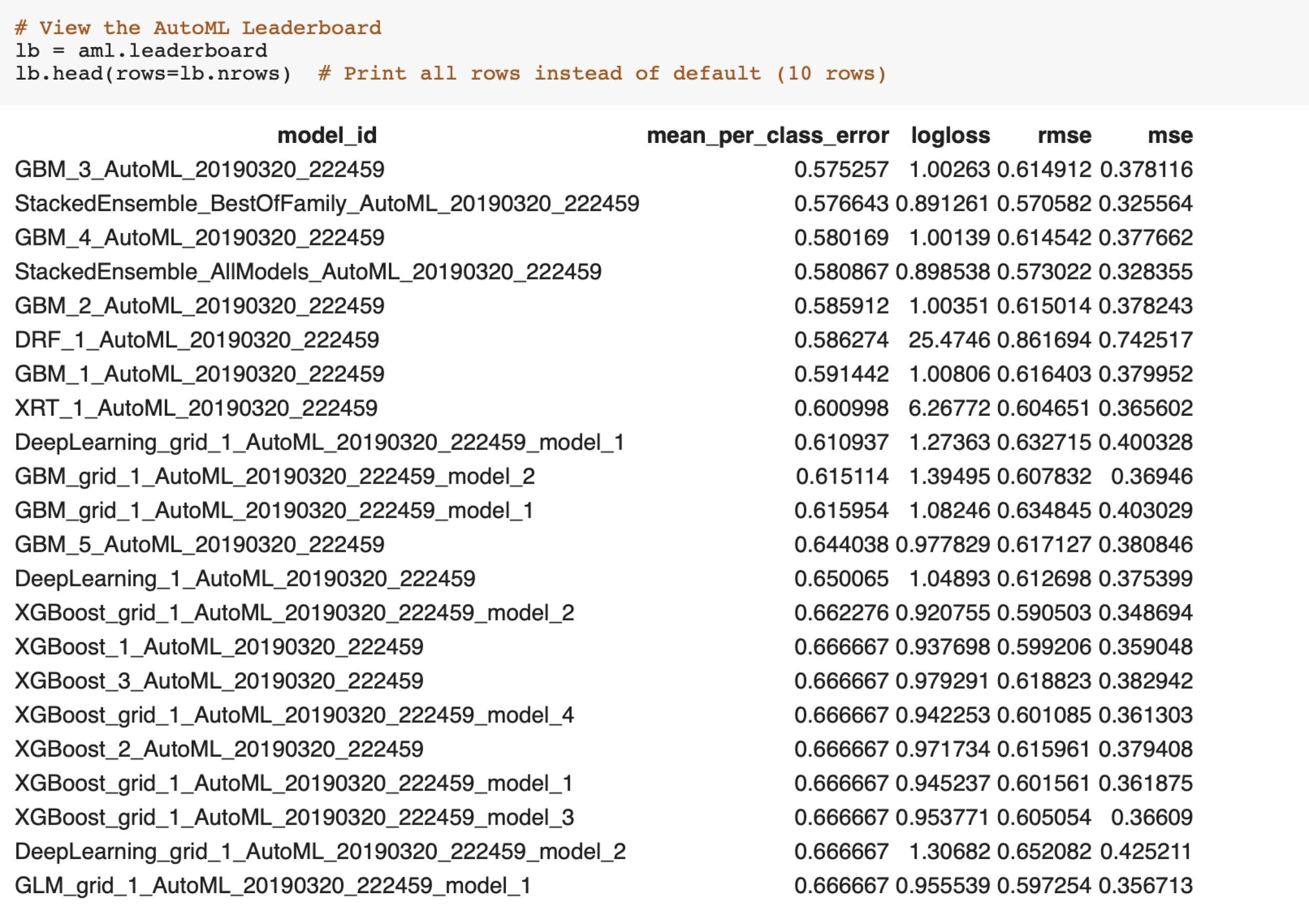


Result:

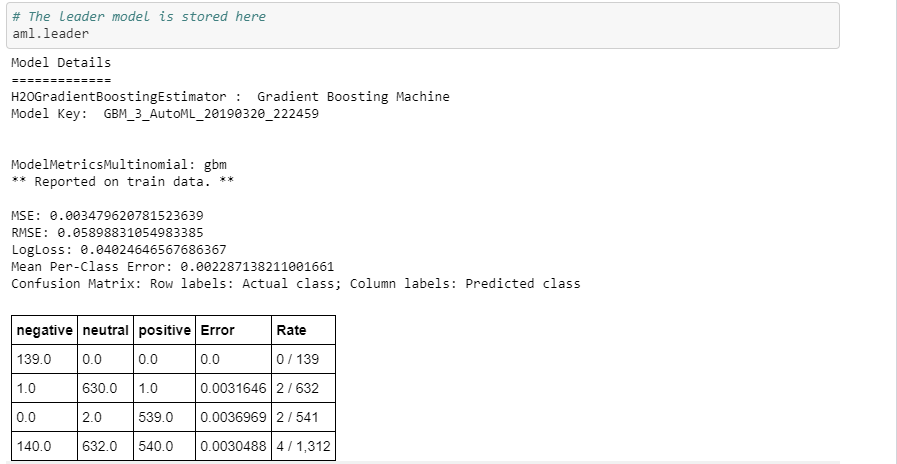


**This model is better than metrics from Experiment 3**

**3. H2o. ai**

****

Best model:



**This model is better than metrics from Experiment 3**

### **Conclusion:**

Our best model so far for auto Machine Learning is using TPO, accuracy is 0.8644