In [ ]:

```
import re
import spacy
import csv
import numpy as np
from sklearn.linear_model import LogisticRegression


adjectives = []

maxent = LogisticRegression(solver="sag")
```

In [ ]:

```
#Load in adjectives
def loadAdj():

    with open("adjectiveList.txt", "r", encoding="utf-8") as fs:

        for line in fs:

            cur = line.split(",")
            for adj in cur:

                if(adj not in adjectives):
                    adjectives.append(adj)

    print("Total Unique Adjectives",len(adjectives))

loadAdj()
```

```
Total Unique Adjectives 18729
```

In [ ]:

```
#Feature Defenitions
def review_len(rvw):

    sen = re.split(" ", rvw)
    return len(sen)

def adj_count(rvw):

    adjs = 0
    adjList = []
    eng = spacy.load("en_core_web_sm")

    sens = re.split(" ", rvw)
    for word in sens:

        if(word in adjectives):
            spacyWord = eng(word)
            # if(spacyWord[0].pos_ == "ADJ"):
            adjList.append(word)
            adjs += 1

    return adjs, adjList
```

In [ ]:

```
rvws = []
labels = []
```

```
#Load in reviews and labels
with open("../review.csv", "r", encoding="utf-8") as csv_file:
    csv_reader = csv.reader(csv_file)
    line = 0

    for row in csv_reader:
        if(line == 0):
            line += 1
        else:
            rvws.append(row[0])
            labels.append(row[1])
            line += 1

    print(line - 1, " reviews loaded")
```

```
300000  reviews loaded
```

In [ ]:

```
train_weights = []
train_rvws = rvws[:35000]
train_labels = labels[:35000]

#Generate weights
for r in train_rvws:

    temp = []
    ads, adlist = adj_count(r)
    l = review_len(r)

    adj_tuple = tuple(adlist)
    temp.append(hash(adj_tuple))
    temp.append(ads)
    temp.append(l)

    train_weights.append(np.array(temp))


maxent.fit(np.array(train_weights), np.array(train_labels))
print("MaxEnt model trained")
```

```
MaxEnt model trained
```

In [ ]:

```
from sklearn import metrics as m

dev_weights = []
dev_eval_labels = labels[35000:50000]
dev_rvws = rvws[35000:50000]

for r in dev_rvws:

    temp = []
    ads, adlist = adj_count(r)
    l = review_len(r)

    adj_tuple = tuple(adlist)
    temp.append(hash(adj_tuple))
    temp.append(ads)
    temp.append(l)
    dev_weights.append(np.array(temp))


predictions = maxent.predict(dev_weights)

P = m.precision_score(y_true=dev_eval_labels, y_pred=predictions, average="micro")
fp = '{0:.4g}'.format(P)
R = m.recall_score(y_true=dev_eval_labels, y_pred=predictions, average="micro")
fr = '{0:.4g}'.format(R)
F1 = m.f1_score(y_true=dev_eval_labels, y_pred=predictions, average="micro")
ff1 = '{0:.4g}'.format(F1)
print("Dev set evaluation:")
print("Micro Scale: \nPrecision = ", fp, " | Recall = ", fr, " | F1-Score = ", ff1)
```

```
Dev set evaluation:
Micro Scale: 
Precision =  0.2582  | Recall =  0.2582  | F1-Score =  0.2582
```