06_07_hw_exemplar

February 13, 2020

0.1 Homework 6+7. MNB and SVMs

```
In [1]: import pandas as pd
        import nltk
        import numpy as np
        import seaborn as sns
        import matplotlib.pyplot as plt
        from tinytext import tinytext
```

```
In [2]: #plt.rcParams['figure.figsize'] = [12, 6]

        #First, lets load and inspect the data

        sourceFile = "deception_data_converted_final.tsv"
        prodReviews = pd.read_table(sourceFile, delimiter="\t")
```

```
In [3]: #pd.set_option('display.max_colwidth', 150)
        print(prodReviews.shape)
        prodReviews.head()
```

```
(92, 3)
```

Out[3]:

	lie	sentiment	review
0	f	n	'Mike\'s Pizza High Point, NY Service was very...
1	f	n	'i really like this buffet restaurant in Marsh...
2	f	n	'After I went shopping with some of my friend,...
3	f	n	'Olive Oil Garden was very disappointing. I ex...
4	f	n	'The Seven Heaven restaurant was never known f...

0.2 Clean up reviews

Here we apply some simple cleaning of the text, and inspect the results

```
In [4]: prodReviews['cleanedReview'] = prodReviews['review'].str.replace("\\","")
        prodReviews['cleanedReview'] = prodReviews['cleanedReview'].str.replace("\'","")
        prodReviews['cleanedReview'] = prodReviews['cleanedReview'].str.replace("\t"," ")
        prodReviews['cleanedReview'] = prodReviews['cleanedReview'].str.replace("^\s+","")
```

```
prodReviews['cleanedReview'] = prodReviews['cleanedReview'].str.replace("\,","")
prodReviews['cleanedReview'] = prodReviews['cleanedReview'].str.replace("\s\d+\s"," ")
prodReviews['cleanedReview'] = prodReviews['cleanedReview'].str.replace("\.","")
prodReviews['cleanedReview'] = prodReviews['cleanedReview'].str.lower()
prodReviews
```

Out[4]:

	lie	sentiment	review \
0	f	n	'Mike\'s Pizza High Point, NY Service was very...
1	f	n	'i really like this buffet restaurant in Marsh...
2	f	n	'After I went shopping with some of my friend,...
3	f	n	'Olive Oil Garden was very disappointing. I ex...
4	f	n	'The Seven Heaven restaurant was never known f...
5	f	n	'I went to XYZ restaurant and had a terrible e...
6	f	n	'I went to ABC restaurant two days ago and I h...
7	f	n	'I went to the Chilis on Erie Blvd and had the...
8	f	n	'OMG. This restaurant is horrible. The recepti...
9	f	n	'Yesterday, I went to a casino-restaurant call...
10	f	n	'Last weekend I went to a place called Ratasti...
11	f	n	'I once went to Chipotle at Marshall Street to...
12	f	n	'I entered the restaurant and a waitress came ...
13	f	n	'Carlo\'s Plate Shack was the worst dining exp...
14	f	n	'This is the last place you would want to dine...
15	f	n	'In each of the diner dish there are at least ...
16	f	n	'I went there with two friends at 6pm. Long qu...
17	f	n	'I had heard that Panera bread is a good place...
18	f	n	'The service was way below average and we had ...
19	f	n	'This place called Samarkand near to SU Main C...
20	f	n	'Usually, I use Yelp to find restaurant. The Y...
21	f	n	'I don\'t usually write reviews on TripAdvisor...
22	f	n	'I recently ate at a restaurant called White C...
23	t	n	'Pizza Hut Syracuse, NY The only thing worth g...
24	t	n	'the staff at this restaurant is very unfriend...
25	t	n	'Friday is the worse restaurant I have ever go...
26	t	n	'This diner was not at all up to par. I\'ve be...
27	t	n	'The worst restaurant I have ever been to ende...
28	t	n	'I went to this restaurant where I had ordered...
29	t	n	'I went to XYZ restaurant last week and I was ...
..	..	...	...
62	f	p	'This Japanese restaurant is so popular recent...
63	f	p	'Hibachi the grill is one of my favorite resta...
64	f	p	'Gannons Isle Ice Cream served the best ice c...
65	f	p	'RIM KAAP One of the best Thai restaurants in ...
66	f	p	'It is a France restaurant, which has Michelin...
67	f	p	'It\'s hard to pick a favorite dining experien...
68	f	p	'I ate at this restaurant called Banana Leaf. ...
69	t	p	'Twin Trees Cicero, NY HUGE salad bar and high...
70	t	p	'i really like this one chicken wings restaura...
71	t	p	'Ruby Tuesday is my favorite America Style Res...

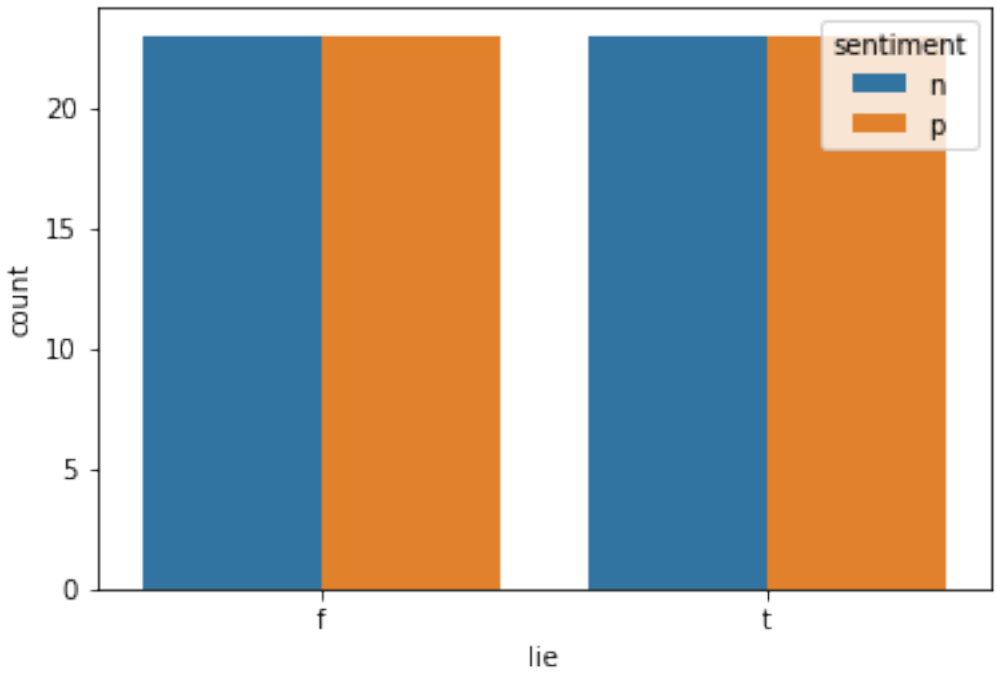
```
72   t         p  'Stronghearts cafe is the BEST! The owners hav...
73   t         p  'The best restaurant I have ever been was a sm...
74   t         p  'My best restaurant is Amer Palace Hotel in my...
75   t         p  'When you walk into TYU you may not have the h...
76   t         p  'I went to this awesome restaurant in San Fran...
77   t         p  'This cafe is located on the 2nd street from t...
78   t         p  'I went to cruise dinner in NYC with Spirit Cr...
79   t         p  'Halo\'s is home. I have been here numerous ti...
80   t         p  'The best restaurant I have gone to is when I ...
81   t         p  'The restaurant looked pretty good, the people...
82   t         p                                                 ?
83   t         p                                                 ?
84   t         p  'A big piano is in the middle of the lobby and...
85   t         p  'Can\'t say too much about it. Just, try it bu...
86   t         p  'Blue Monkey Cafe is my favorite Japanese rest...
87   t         p  'Pastablities is a locally owned restaurant in...
88   t         p  'I like the Pizza at Dominoes for their specia...
89   t         p  'It was a really amazing Japanese restaurant. ...
90   t         p  'How do I even pick a best experience at Joe\'...
91   t         p  'My sister and I ate at this restaurant called...
```

```
                                          cleanedReview
0   mikes pizza high point ny service was very slo...
1   i really like this buffet restaurant in marsha...
2   after i went shopping with some of my friend w...
3   olive oil garden was very disappointing i expe...
4   the seven heaven restaurant was never known fo...
5   i went to xyz restaurant and had a terrible ex...
6   i went to abc restaurant two days ago and i ha...
7   i went to the chilis on erie blvd and had the ...
8   omg this restaurant is horrible the receptioni...
9   yesterday i went to a casino-restaurant called...
10  last weekend i went to a place called ratastic...
11  i once went to chipotle at marshall street to ...
12  i entered the restaurant and a waitress came b...
13  carlos plate shack was the worst dining experi...
14  this is the last place you would want to dine ...
15  in each of the diner dish there are at least o...
16  i went there with two friends at 6pm long queu...
17  i had heard that panera bread is a good place ...
18  the service was way below average and we had t...
19  this place called samarkand near to su main ca...
20  usually i use yelp to find restaurant the yelp...
21  i dont usually write reviews on tripadvisor bu...
22  i recently ate at a restaurant called white ca...
23  pizza hut syracuse ny the only thing worth goi...
24  the staff at this restaurant is very unfriendl...
25  friday is the worse restaurant i have ever gon...
```

```
26  this diner was not at all up to par ive been t...
27  the worst restaurant i have ever been to ended...
28  i went to this restaurant where i had ordered ...
29  i went to xyz restaurant last week and i was v...
..                                                ...
62  this japanese restaurant is so popular recentl...
63  hibachi the grill is one of my favorite restau...
64  gannons isle ice cream served the best ice cr...
65  rim kaap one of the best thai restaurants in t...
66  it is a france restaurant which has michelin t...
67  its hard to pick a favorite dining experience ...
68  i ate at this restaurant called banana leaf as...
69  twin trees cicero ny huge salad bar and high q...
70  i really like this one chicken wings restauran...
71  ruby tuesday is my favorite america style rest...
72  stronghearts cafe is the best! the owners have...
73  the best restaurant i have ever been was a sma...
74  my best restaurant is amer palace hotel in my ...
75  when you walk into tyu you may not have the hi...
76  i went to this awesome restaurant in san franc...
77  this cafe is located on the 2nd street from th...
78  i went to cruise dinner in nyc with spirit cru...
79  halos is home i have been here numerous times ...
80  the best restaurant i have gone to is when i w...
81  the restaurant looked pretty good the people a...
82                                                 ?
83                                                 ?
84  a big piano is in the middle of the lobby and ...
85  cant say too much about it just try it buddy!!...
86  blue monkey cafe is my favorite japanese resta...
87  pastablities is a locally owned restaurant in ...
88  i like the pizza at dominoes for their special...
89  it was a really amazing japanese restaurant th...
90  how do i even pick a best experience at joes n...
91  my sister and i ate at this restaurant called ...

[92 rows x 4 columns]
```

In [5]:
```
sns.countplot(data=prodReviews, x="lie", hue='sentiment')
# The reviews are evenly distributed across lies and sentiment -- GREAT!
```

Out[5]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x204dbb68860>
```

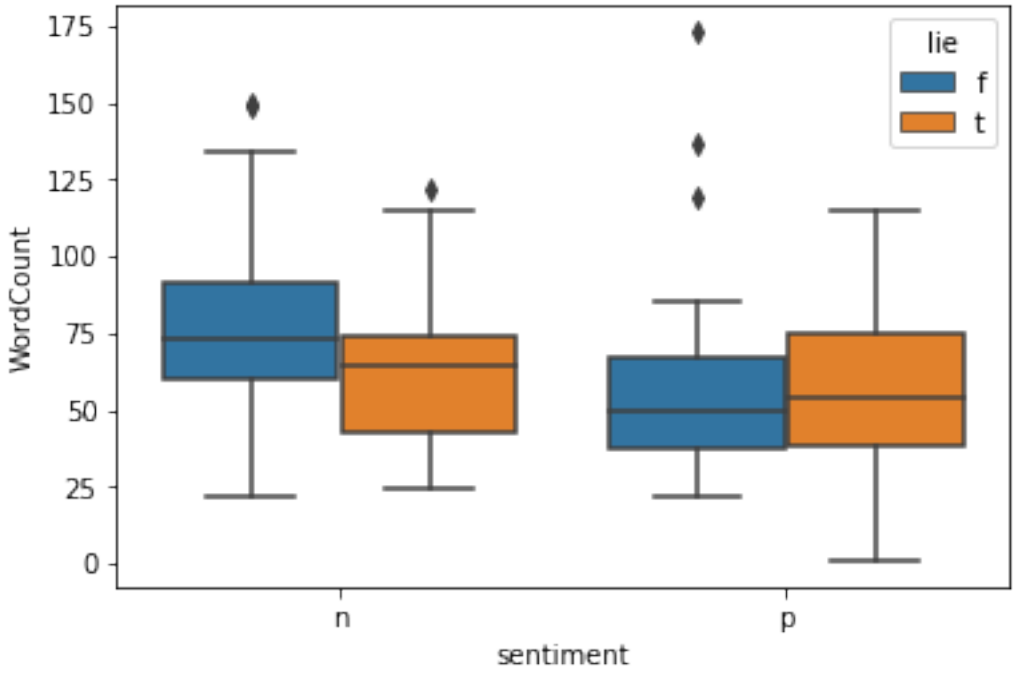


In [6]:
```
# Lets investigate the wordcounts.

prodReviews['WordCount'] = prodReviews['cleanedReview'].str.count("\S\s+\S")+1
prodReviews['WordCount'].describe()
```

Out[6]:
```
count     92.000000
mean      71.184783
std       55.263391
min        1.000000
25%       43.000000
50%       63.000000
75%       78.750000
max      458.000000
Name: WordCount, dtype: float64
```

In [7]:
```
# Remove the outliers when comparing the impact of word count
sns.boxplot(data=prodReviews[prodReviews['WordCount'] < 200], x='sentiment', y='WordCou
```

Out[7]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x204e0f30588>
```




```
In [8]: import wordcloud
        from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
        from nltk.corpus import stopwords

In [9]: allReviews = " ".join(rv for rv in prodReviews['cleanedReview'])

        # Next for EDA, lets investigate word frequency. Create a word cloud to help visualize

        wordcloud = WordCloud(background_color='black', max_words=100, stopwords=stopwords.word
        plt.imshow(wordcloud, interpolation='bilinear')
        plt.title("All Reviews")
        plt.axis("off")
        plt.show()
```

0.3 Investigate stop words for the lie detection model

In our experiments, we investigate the effect of stopwords. Specifically we will compare results using no stopwords, standard English Stop Words, and Custom Created Stopwords.

In [10]:
```
## Define the stop words

lieStopWords = stopwords.words('english')
# Remmove most common words across categories
lieStopWords.extend(["restaurant","food","place","went","would"])
lieStopWords.extend(["friends","friend","like","good","best","great","service","ordere
##Remove food types
lieStopWords.extend(["chicken","pizza","chinese","japanese","salad","iced","tea","sush
## Remove common restaurant nouns
lieStopWords.extend(["dining","dinner","dine","dish","dishes","waiters","waiter","wait
## Remove generic concepts
lieStopWords.extend(["experience","order","really"])
#Remove other key words common across categories
lieStopWords.extend(["never","one","served","price","staff","back","delicious","flavor
lieStopWords.extend(["hour","last","minutes","made","need","overall","prices","table"

## Preview the stop words in word clouds

prodReviewsTrue = prodReviews[prodReviews['lie'] =='t']
trueReviews = " ".join(rv for rv in prodReviewsTrue['cleanedReview'])

wordcloud = WordCloud(background_color='black', max_words=100, stopwords=lieStopWords)
plt.imshow(wordcloud, interpolation='bilinear')
```

```
plt.title("Truthful Reviews")
plt.axis("off")
plt.show()

prodReviewsFalse = prodReviews[prodReviews['lie'] =='f']
falseReviews = " ".join(rv for rv in prodReviewsFalse['cleanedReview'])

wordcloud = WordCloud(background_color='black', max_words=100, stopwords=lieStopWords)
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Fake Reviews")
plt.axis("off")
plt.show()
```

Truthful Reviews

Fake Reviews

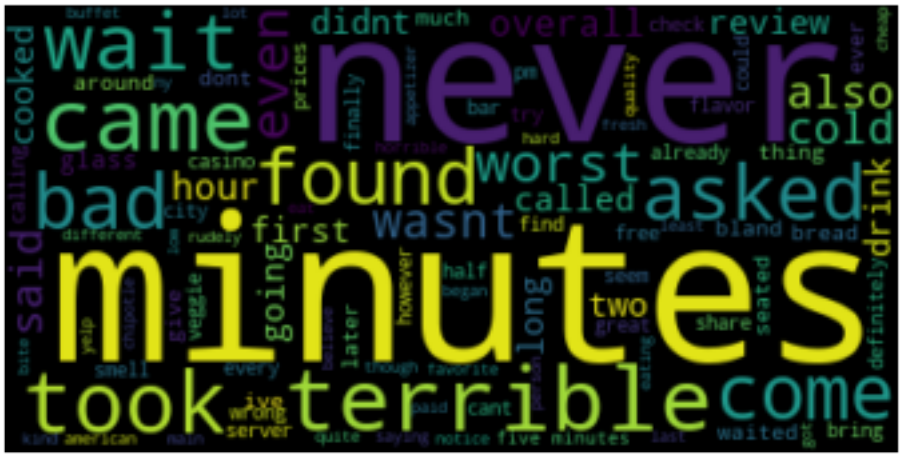
0.4 Stop words for the sentiment identification model

In our experiments, we investigate the effect of stopwords. Specifically we will compare results using no stopwords, standard English Stop Words, and Custom Created Stopwords.

In [11]:
```
## Define the stop words

sentStopWords = stopwords.words('english')
# Remmove most common words across categories
sentStopWords.extend(["restaurant","restaurants","food","place","went"])
sentStopWords.extend(["good","like","go"])
sentStopWords.extend(["service","order","ordered","us","friend","friends","would"])
sentStopWords.extend(["back","experience","life","taste","time","serve","served"])
## Remove common restaurant nouns
sentStopWords.extend(["dining","dinner","dine","dish","dishes","waiters","waiter","wa
##Remove food types
sentStopWords.extend(["chicken","pizza","chinese","japanese","salad","iced","tea","su
## Remove additional common words
sentStopWords.extend(["get","little","made","meal","one","people","plate","price","re



## Preview the stop words in word clouds

prodReviewsPositive = prodReviews[prodReviews['sentiment'] =='p']
posReviews = " ".join(rv for rv in prodReviewsPositive['cleanedReview'])

wordcloud = WordCloud(background_color='black', max_words=100, stopwords=sentStopWords
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Positive Reviews")
plt.axis("off")
plt.show()

prodReviewsNegative = prodReviews[prodReviews['sentiment'] =='n']
negativeReviews = " ".join(rv for rv in prodReviewsNegative['cleanedReview'])

wordcloud = WordCloud(background_color='black', max_words=100, stopwords=sentStopWords
plt.imshow(wordcloud, interpolation='bilinear')
plt.title("Negative Reviews")
plt.axis("off")
plt.show()
```

Positive Reviews

Negative Reviews



0.5 Vectorize the Data.

Below we identify frequent words, apply stemmer, and vectorize results.

```
In [12]: from nltk.tokenize import TreebankWordTokenizer
         from nltk.stem import SnowballStemmer
         stemmer = SnowballStemmer("english")

         allReviewTokens = TreebankWordTokenizer().tokenize(allReviews)
```

```
        lieReviewTokens = []
        lieReviewStemmedTokens = []
        sentReviewTokens = []
        sentReviewStemmedTokens = []
        for tk in allReviewTokens:
            stemmedTk = stemmer.stem(tk)
            if lieStopWords.count(tk) == 0:
                lieReviewTokens.append(tk)
                if (lieStopWords.count(stemmedTk) == 0):
                    lieReviewStemmedTokens.append(stemmedTk)
            if sentStopWords.count(tk) == 0:
                sentReviewTokens.append(tk)
                if (sentStopWords.count(stemmedTk) == 0):
                    sentReviewStemmedTokens.append(stemmedTk)

        print("Most Frequent Words for Lie Detection")
        lieReviewFreq = nltk.FreqDist(lieReviewTokens)
        print(lieReviewFreq.most_common(50))
        print()
        print("Most Frequent Words for Lie Detection (Stemmed)")
        lieReviewFreq = nltk.FreqDist(lieReviewStemmedTokens)
        print(lieReviewFreq.most_common(50))
        print()
        print("Most Frequent Words for Sentiment 'Identification'")
        sentReviewFreq = nltk.FreqDist(sentReviewTokens)
        print(sentReviewFreq.most_common(50))
        print()
        print("Most Frequent Words for Sentiment Identification (Stemmed)")
        sentReviewFreq = nltk.FreqDist(sentReviewStemmedTokens)
        print(sentReviewFreq.most_common(50))
```

```
Most Frequent Words for Lie Detection
[('!', 152), ('(', 19), (')', 18), (':', 16), ('plate', 15), ('time', 14), ('amazing', 14), ('

Most Frequent Words for Lie Detection (Stemmed)
[('!', 152), ('(', 19), (')', 18), ('time', 18), ('call', 17), ('plate', 16), (':', 16), ('ask

Most Frequent Words for Sentiment 'Identification'
[('!', 152), ('best', 31), ('great', 24), ('minutes', 23), ('(', 19), (')', 18), ('never', 16)

Most Frequent Words for Sentiment Identification (Stemmed)
[('!', 152), ('best', 31), ('great', 24), ('minut', 23), ('(', 19), ('wait', 19), (')', 18), (
```

```
In [13]: def stemText(txt):
             tkns = TreebankWordTokenizer().tokenize(txt)
             newTkns = []
             for t in tkns:
```

```
            newTkns.append(stemmer.stem(t))
        newTxt = " ".join(newTkns)
        return newTxt

    stemText("life's like a bag of chocolates")
```

Out[13]: "life 's like a bag of chocol"

```
In [14]: prodReviews['stemmedReview'] = prodReviews['cleanedReview'].apply(stemText)

    modelDf = pd.DataFrame(prodReviews.drop(columns=['review','cleanedReview','stemmedRevi
    #modelDf['isPositive'] = modelDf['sentiment'].mask(modelDf['sentiment'] == 'n',0)
    #modelDf['isPositive'] = modelDf['isPositive'].mask(modelDf['isPositive'] == 'p',1)
    #modelDf = modelDf.drop(columns=['sentiment'])
    modelDf.head()
```

Out[14]:

	lie	sentiment	WordCount
0	f	n	43
1	f	n	58
2	f	n	22
3	f	n	41
4	f	n	63

```
In [15]: from sklearn.feature_extraction.text import CountVectorizer
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.naive_bayes import MultinomialNB
    from sklearn.naive_bayes import BernoulliNB
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import confusion_matrix

In [16]: def getFeatureVector(vectorizer, wordList, modelDf, columnToDrop):
        featureVector = vectorizer.fit_transform(wordList)
        featureVectorDf = pd.DataFrame(columns=vectorizer.get_feature_names(), data=featu
        modelWithFeatureDf = modelDf.join(featureVectorDf)
        modelWithFeatureVector = modelWithFeatureDf.drop(columns=["WordCount", columnToDr
        return modelWithFeatureVector

    def getCountVectorizer(binary=False, ngram=(1,1), minDocFreq=.03, maxDocFreq=.4, maxF
        if (stopwords == None):
            return CountVectorizer(encoding='latin-1', binary=True, ngram_range=ngram, mi
        else:
            return CountVectorizer(encoding='latin-1', binary=True, ngram_range=ngram, mi

    def getTfidfVectorizer(ngram=(1,1), minDocFreq=.03, maxDocFreq=.4, maxFeat=50, sw=Non
        if (stopwords == None):
            return TfidfVectorizer(encoding='latin-1', use_idf=True, ngram_range=ngram, m
        else:
            return TfidfVectorizer(encoding='latin-1', use_idf=True, ngram_range=ngram, m

In [17]: #testVector = getFeatureVector(sent_bool_vectorizer, prodReviews['cleanedReview'].val
    #testVector.head()
```

0.6 Experimental Design

MNB results are compared using the different preprocessed data sets for a robust experimental design. Training and Test Sets are separated, and average results are presented in tabular form

In [18]:
```
from sklearn.model_selection import cross_validate

def runNBModel(vect, labelName, labelValues, model="multi"):
    vectT = vect[vect[labelName] == labelValues[0]]
    vectF = vect[vect[labelName] == labelValues[1]]
    train_xT, test_xT, train_yT, test_yT = train_test_split(vectT.drop(columns=[labell
    train_xF, test_xF, train_yF, test_yF = train_test_split(vectF.drop(columns=[labell

    train_x = train_xT.append(train_xF)
    train_y = train_yT.append(train_yF)
    test_x = test_xT.append(test_xF)
    test_y = test_yT.append(test_yF)

    if (model == "multi"):
        nbModel = MultinomialNB()
    elif (model == "bern"):
        nbModel = BernoulliNB()
    nbModel.fit(train_x, train_y)
    pred = nbModel.predict(test_x)
    score = nbModel.score(test_x, test_y)
    #print("Accuracy: ", score)
    #print("Predictions: ", pred)
    #print("Confusion Matrix: \n", confusion_matrix(test_y, pred, labels=labelValues),
    #print()
    return score

def runNBModelCv(vect, labelName, labelValues, folds=5, model="multi", verbose=True):

    if (model == "multi"):
        nbModel = MultinomialNB()
    elif (model == "bern"):
        nbModel = BernoulliNB()
    cv_results = cross_validate(nbModel, vect.drop(columns=[labelName]), vect[labelNa
    #print(cv_results.keys())
    if (verbose):
        print("Train Scores:", cv_results['train_score'])
        print("\tAverage Train Score:", np.mean(cv_results['train_score']))
        print("Test Scores:", cv_results['test_score'])
        print("\tAverage Test Score:", np.mean(cv_results['test_score']))
    return (np.mean(cv_results['test_score']), cv_results['test_score'])
```

In [19]:
```
timesToRunModel = 10
whichReviews = prodReviews['cleanedReview'].values
verbose = False
```

```
vectorMapList = []
## lets define some parameters for the lie detection label
vectorMapList.append(("Lie","Bool No Stop",getFeatureVector(getCountVectorizer(), whi
vectorMapList.append(("Lie","Bool Eng Stop",getFeatureVector(getCountVectorizer(sw=sto
vectorMapList.append(("Lie","Bool Cust Stop 50",getFeatureVector(getCountVectorizer(sw
vectorMapList.append(("Lie","Bool Cust Stop 200",getFeatureVector(getCountVectorizer(s
vectorMapList.append(("Lie","Bool Cust Stop 200 / .02-.6",getFeatureVector(getCountVec

vectorMapList.append(("Lie","Count Cust Stop 200",getFeatureVector(getCountVectorizer
vectorMapList.append(("Lie","Tfidf Cust Stop 200",getFeatureVector(getTfidfVectorizer
vectorMapList.append(("Lie","Tfidf Cust Stop 200 - Stemmed",getFeatureVector(getTfidfV
vectorMapList.append(("Lie","Tfidf Cust Stop 200 / .02-.6",getFeatureVector(getTfidfVe

## Now lets define some parameters for the sentiment analysis label
vectorMapList.append(("Sentiment","Bool No Stop",getFeatureVector(getCountVectorizer()
vectorMapList.append(("Sentiment","Bool Eng Stop",getFeatureVector(getCountVectorizer
vectorMapList.append(("Sentiment","Bool Cust Stop 50",getFeatureVector(getCountVectori
vectorMapList.append(("Sentiment","Bool Cust Stop 50 - Stemmed",getFeatureVector(getCo
vectorMapList.append(("Sentiment","Bool Cust Stop 200",getFeatureVector(getCountVector
vectorMapList.append(("Sentiment","Bool Cust Stop 200 / .02-.6",getFeatureVector(getCo

vectorMapList.append(("Sentiment","Count Cust Stop 200",getFeatureVector(getCountVecto
vectorMapList.append(("Sentiment","Tfidf Cust Stop 200",getFeatureVector(getTfidfVecto
#vectorMapList.append(("Sentiment","Tfidf Cust Stop 200 - Stemmed",getFeatureVector(g
vectorMapList.append(("Sentiment","Tfidf Cust Stop 200 / .02-.6",getFeatureVector(getT


lieCategories = ["t","f"]
sentimentCategories = ["p","n"]
averageScore = pd.DataFrame(columns=["category", "model", "test.scores", "test.average
for (c, l, v, m) in vectorMapList:
    scores = []
    for n in range(timesToRunModel):
        if (c == 'Lie'):
            scores.append(runNBModel(v, "lie", lieCategories, m))
        elif (c == 'Sentiment'):
            scores.append(runNBModel(v, "sentiment", sentimentCategories, m))

    testAvg = np.mean(scores)

    if (verbose):
        print()
        print("**********************")
        print("**", c, "-", l, " Vectorizer")
        print("**********************")
        print(scores)
        print(testAvg)
```

```
            print()

        if (c == 'Lie'):
            (cvTestAvg, cvTestScores) = runNBModelCv(v, "lie", lieCategories, timesToRunM
        elif(c == 'Sentiment'):
            (cvTestAvg, cvTestScores) = runNBModelCv(v, "sentiment", sentimentCategories,

        tmpDf = pd.DataFrame(columns=["category", "model", "test.scores", "test.average",
        tmpDf['category'] = [c]
        tmpDf['model'] = [l]
        tmpDf['test.scores'] = [scores]
        tmpDf['test.average'] = [testAvg]
        tmpDf['cv.test.scores'] = [cvTestScores]
        tmpDf['cv.test.average'] = [cvTestAvg]
        tmpDf['features'] = [len(v.columns)]
        averageScore = averageScore.append(tmpDf)

    if (verbose == False):
        averageScore = averageScore.drop(columns=["test.scores",'cv.test.scores'])

    averageScore
```

Out[19]:

	category	model	test.average	cv.test.average	features
0	Lie	Bool No Stop	0.43	0.4900	51
0	Lie	Bool Eng Stop	0.54	0.5775	51
0	Lie	Bool Cust Stop 50	0.56	0.6275	51
0	Lie	Bool Cust Stop 200	0.54	0.6075	201
0	Lie	Bool Cust Stop 200 / .02-.6	0.61	0.6075	201
0	Lie	Count Cust Stop 200	0.59	0.6175	
0	Lie	Tfidf Cust Stop 200	0.61	0.6500	
0	Lie	Tfidf Cust Stop 200 - Stemmed	0.57	0.6375	
0	Lie	Tfidf Cust Stop 200 / .02-.6	0.67	0.6300	
0	Sentiment	Bool No Stop	0.75	0.6550	
0	Sentiment	Bool Eng Stop	0.81	0.7800	
0	Sentiment	Bool Cust Stop 50	0.80	0.8775	
0	Sentiment	Bool Cust Stop 50 - Stemmed	0.84	0.8200	
0	Sentiment	Bool Cust Stop 200	0.81	0.8250	
0	Sentiment	Bool Cust Stop 200 / .02-.6	0.80	0.7900	
0	Sentiment	Count Cust Stop 200	0.79	0.8350	
0	Sentiment	Tfidf Cust Stop 200	0.82	0.8475	
0	Sentiment	Tfidf Cust Stop 200 / .02-.6	0.82	0.8275	

```
0      201
0      201
0      201
0      201
0       51
0       51
0       51
0       51
0      201
0      201
0      201
0      201
0      201
```

```
In [20]: sns.boxplot(data=averageScore, y='cv.test.average', x='category')
```

```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x204e2236be0>
```