**import** pandas **as** pd

**import** preprocess

**import** numpy **as** np

*# Read in labeled stock sentiment data*

data **=** pd**.**read\_csv('train/stock\_data.csv')

*# Preprocess tweets for NLP analysis*

data **=** preprocess**.**Preprocess\_Tweets(data)

display(data)

C:\Users\aj1ro\anaconda3\lib\site-packages\pandas\core\strings\object\_array.py:158: FutureWarning: Possible nested set at position 1

pat = re.compile(pat, flags=flags)

|  | **Text** | **Sentiment** | **Text\_Cleaned** |
| --- | --- | --- | --- |
| **0** | Kickers on my watchlist XIDE TIT SOQ PNK CPW B... | 1 | kickers on my watchlist xide tit soq pnk cpw b... |
| **1** | user: AAP MOVIE. 55% return for the FEA/GEED i... | 1 | aap movie 55 percent return for the fea and ge... |
| **2** | user I'd be afraid to short AMZN - they are lo... | 1 | id be afraid to short amzn to they are looking... |
| **3** | MNTA Over 12.00 | 1 | mnta over 12.00 |
| **4** | OI Over 21.37 | 1 | oi over 21.37 |
| **...** | ... | ... | ... |
| **5786** | Industry body CII said #discoms are likely to ... | -1 | industry body cii said are likely to suffer a ... |
| **5787** | #Gold prices slip below Rs 46,000 as #investor... | -1 | prices slip below rs 46000 as book profits ami... |
| **5788** | Workers at Bajaj Auto have agreed to a 10% wag... | 1 | workers at bajaj auto have agreed to a 10 perc... |
| **5789** | #Sharemarket LIVE: Sensex off day’s high, up 6... | 1 | live sensex off day’s high up 600 points tests... |
| **5790** | #Sensex, #Nifty climb off day's highs, still u... | 1 | climb off days highs still up 2 percent key fa... |

5791 rows × 3 columns

In [2]:

**import** nltk

*#nltk.download('vader\_lexicon')*

**from** nltk.sentiment.vader **import** SentimentIntensityAnalyzer

*# Prepare Vader sentiment analyzer*

sid **=** SentimentIntensityAnalyzer()

*# Predict sentiment with Vader classifier*

data['Vader\_Scores'] **=** data['Text\_Cleaned']**.**apply(**lambda** score: sid**.**polarity\_scores(score)['compound'])

data['Vader\_Prediction'] **=** data['Vader\_Scores']**.**apply(**lambda** score: 1 **if** score **>=**0 **else** **-**1)

*# Print Vader sentiment accuracy*

print('Vader Accuracy:', round((len(data[data['Sentiment']**==**data['Vader\_Prediction']])**/**len(data)) **\***100, 2), '%', '\n')

Accuracy: 66.48 %

In [3]:

**import** nltk

*#nltk.download("stopwords")*

**from** nltk.corpus **import** stopwords

*# Create list of custom stop words to remove*

StopWords **=** set([s**.**replace("'", '') **for** s **in** stopwords**.**words('english') **if** s **not** **in** ['not', 'up', 'down', 'above', 'below', 'under', 'over']])

print(StopWords)

*# Remove stop words for tweets*

data['Text\_Processed'] **=** data['Text\_Cleaned']**.**apply(**lambda** s: " "**.**join([word **for** word **in** s**.**split() **if** word **not** **in** StopWords]))

data['Text\_Processed'] **=** data['Text\_Processed']**.**str**.**strip()

display(data)

{'its', 'which', 'on', 'ma', 'was', 'each', 'you', 'but', 'when', 'them', 'with', 'there', 'only', 'can', 'isnt', 'of', 'ain', 'what', 'mightn', 'am', 'such', 'further', 'shes', 'needn', 'ourselves', 'thatll', 'this', 'through', 'shan', 'until', 'those', 'into', 'her', 'in', 'hadnt', 'wont', 'mightnt', 'mustnt', 'arent', 'out', 'didn', 'than', 'shouldnt', 'and', 'the', 'are', 'himself', 'o', 'after', 'she', 'couldn', 'they', 'my', 'other', 'do', 'few', 'itself', 'youd', 'neednt', 'during', 'won', 'a', 'herself', 'now', 'has', 'shouldn', 'he', 'had', 'off', 'why', 'shouldve', 'wasnt', 'hasn', 'or', 'who', 'having', 'being', 'that', 'be', 'these', 'by', 'we', 'here', 'more', 'so', 'too', 'then', 'mustn', 'aren', 'their', 'where', 'i', 'youre', 'been', 'wouldnt', 'our', 'to', 'couldnt', 'youve', 'no', 'haven', 've', 'isn', 'd', 'don', 'your', 'before', 'at', 'me', 's', 'him', 'his', 'as', 'nor', 'an', 'shant', 'very', 'doesn', 'it', 'yourself', 'all', 'were', 'will', 'if', 'y', 'once', 'should', 'against', 'wouldn', 'yours', 'just', 'about', 'weren', 'does', 'dont', 'wasn', 'myself', 'themselves', 'how', 'from', 'while', 'did', 'yourselves', 'same', 'm', 'again', 'between', 'doing', 'own', 'both', 't', 'havent', 'werent', 'youll', 'hers', 'have', 'ours', 'll', 'any', 'some', 'theirs', 'doesnt', 'because', 'whom', 'most', 'hasnt', 'hadn', 'is', 'didnt', 're', 'for'}

|  | **Text** | **Sentiment** | **Text\_Cleaned** | **Vader\_Scores** | **Vader\_Prediction** | **Text\_Processed** |
| --- | --- | --- | --- | --- | --- | --- |
| **0** | Kickers on my watchlist XIDE TIT SOQ PNK CPW B... | 1 | kickers on my watchlist xide tit soq pnk cpw b... | 0.0000 | 1 | kickers watchlist xide tit soq pnk cpw bpz aj ... |
| **1** | user: AAP MOVIE. 55% return for the FEA/GEED i... | 1 | aap movie 55 percent return for the fea and ge... | 0.6249 | 1 | aap movie 55 percent return fea geed indicator... |
| **2** | user I'd be afraid to short AMZN - they are lo... | 1 | id be afraid to short amzn to they are looking... | 0.3612 | 1 | id afraid short amzn looking like near monopol... |
| **3** | MNTA Over 12.00 | 1 | mnta over 12.00 | 0.0000 | 1 | mnta over 12.00 |
| **4** | OI Over 21.37 | 1 | oi over 21.37 | 0.0000 | 1 | oi over 21.37 |
| **...** | ... | ... | ... | ... | ... | ... |
| **5786** | Industry body CII said #discoms are likely to ... | -1 | industry body cii said are likely to suffer a ... | -0.7003 | -1 | industry body cii said likely suffer net reven... |
| **5787** | #Gold prices slip below Rs 46,000 as #investor... | -1 | prices slip below rs 46000 as book profits ami... | 0.0258 | 1 | prices slip below rs 46000 book profits amid l... |
| **5788** | Workers at Bajaj Auto have agreed to a 10% wag... | 1 | workers at bajaj auto have agreed to a 10 perc... | 0.0000 | 1 | workers bajaj auto agreed 10 percent wage cut ... |
| **5789** | #Sharemarket LIVE: Sensex off day’s high, up 6... | 1 | live sensex off day’s high up 600 points tests... | 0.0000 | 1 | live sensex day’s high up 600 points tests 920... |
| **5790** | #Sensex, #Nifty climb off day's highs, still u... | 1 | climb off days highs still up 2 percent key fa... | 0.0000 | 1 | climb days highs still up 2 percent key factor... |

5791 rows × 6 columns

In [4]:

**from** sklearn.feature\_extraction.text **import** TfidfVectorizer

**from** sklearn.model\_selection **import** StratifiedKFold, cross\_val\_score

**from** sklearn.naive\_bayes **import** MultinomialNB

**import** matplotlib.pyplot **as** plt

*# Split data into 80/20 train-test split*

train\_pct **=** .8

np**.**random**.**seed(1)

idx **=** np**.**random**.**permutation(len(data))

X\_train **=** data['Text\_Processed']**.**values[idx[:int(train\_pct**\***len(data))]]

y\_train **=** data['Sentiment']**.**values[idx[:int(train\_pct**\***len(data))]]

y\_train[y\_train**==-**1] **=** 0

X\_test **=** data['Text\_Processed']**.**values[idx[int(train\_pct**\***len(data)):]]

y\_test **=** data['Sentiment']**.**values[idx[int(train\_pct**\***len(data)):]]

y\_test[y\_test**==-**1] **=** 0

*# Calculate TF-IDF for Naive Bayes classification*

tf\_idf **=** TfidfVectorizer(ngram\_range**=**(1, 3),

binary**=True**,

smooth\_idf**=False**)

*# Get TF-IDF for Train and Test data*

X\_train\_tfidf **=** tf\_idf**.**fit\_transform(X\_train)

X\_test\_tfidf **=** tf\_idf**.**transform(X\_test)

*# Define function to determine accuracy of model*

**def** get\_auc\_CV(model):

*# Set KFold to shuffle data before the split*

kf **=** StratifiedKFold(5, shuffle**=True**, random\_state**=**1)

*# Get AUC scores*

auc **=** cross\_val\_score(model, X\_train\_tfidf, y\_train, scoring**=**"roc\_auc", cv**=**kf)

**return** auc**.**mean()

*# Initialize models with multiple alpha values to find best model*

alphas **=** np**.**arange(1,10,0.1)

models **=** [MultinomialNB(alpha**=**i) **for** i **in** alphas]

*# Find best performing model*

accs **=** []

**for** model **in** models:

accs**.**append(get\_auc\_CV(model))

accs **=** np**.**array(accs)

*# Get best performing alpha value to continue with*

best\_alpha **=** round(alphas[accs**.**argmax()], 1)

*# Print best alpha value and accuracy*

print('Best alpha: ', best\_alpha, ' | Best Score: ', round(accs**.**max()**\***100, 2))

*# Plot accuracies per alpha values*

plt**.**figure(figsize**=**(15,7))

plt**.**plot(alphas, accs)

plt**.**show()

*# Retrain best performing model*

best\_model **=** MultinomialNB(alpha**=**best\_alpha)

best\_model**.**fit(X\_train\_tfidf, y\_train)

*# Predict test data with best model*

probs **=** best\_model**.**predict\_proba(X\_test\_tfidf)

*# Print accuracy of best performing model on tweet sentiment analysis*

print('Naive-Bayes Accuracy:', round(len(np**.**where(y\_test **==** probs**.**argmax(axis**=**1))[0])**/**len(probs) **\*** 100, 2), '%')

Best alpha: 1.0 | Best Score: 83.04

Accuracy: 66.78 %