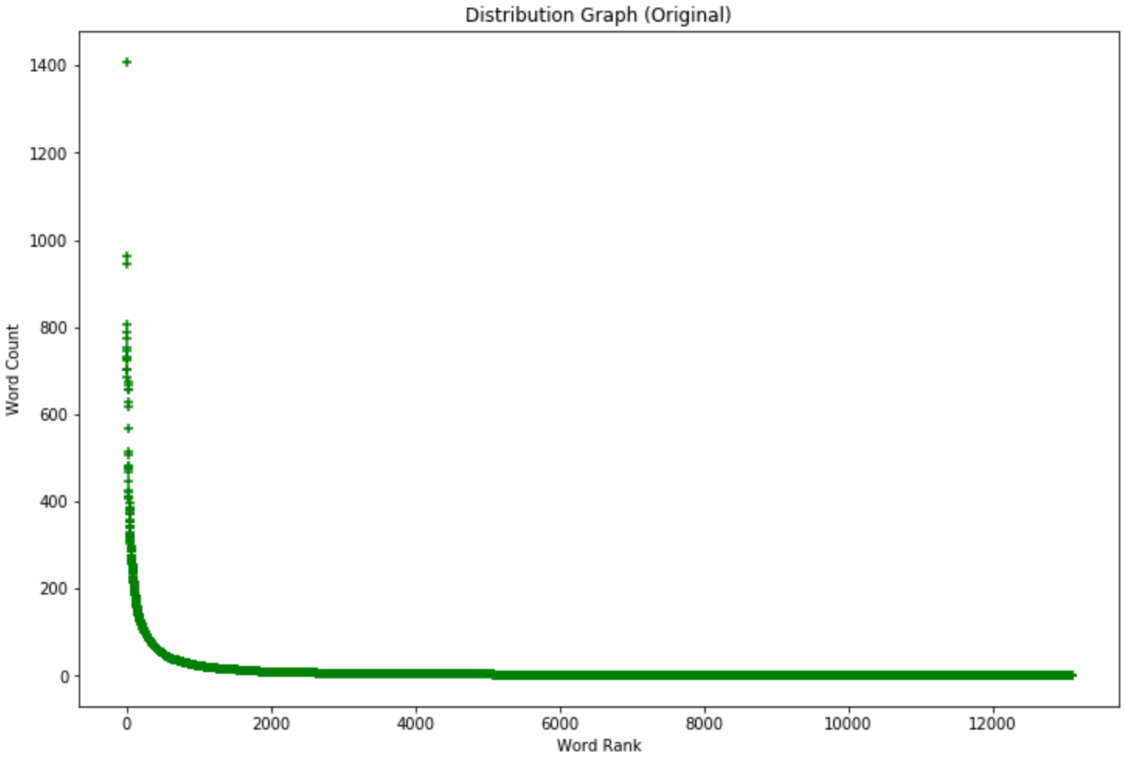
1 Distribution Graph (Words Count Vs Word Rank)

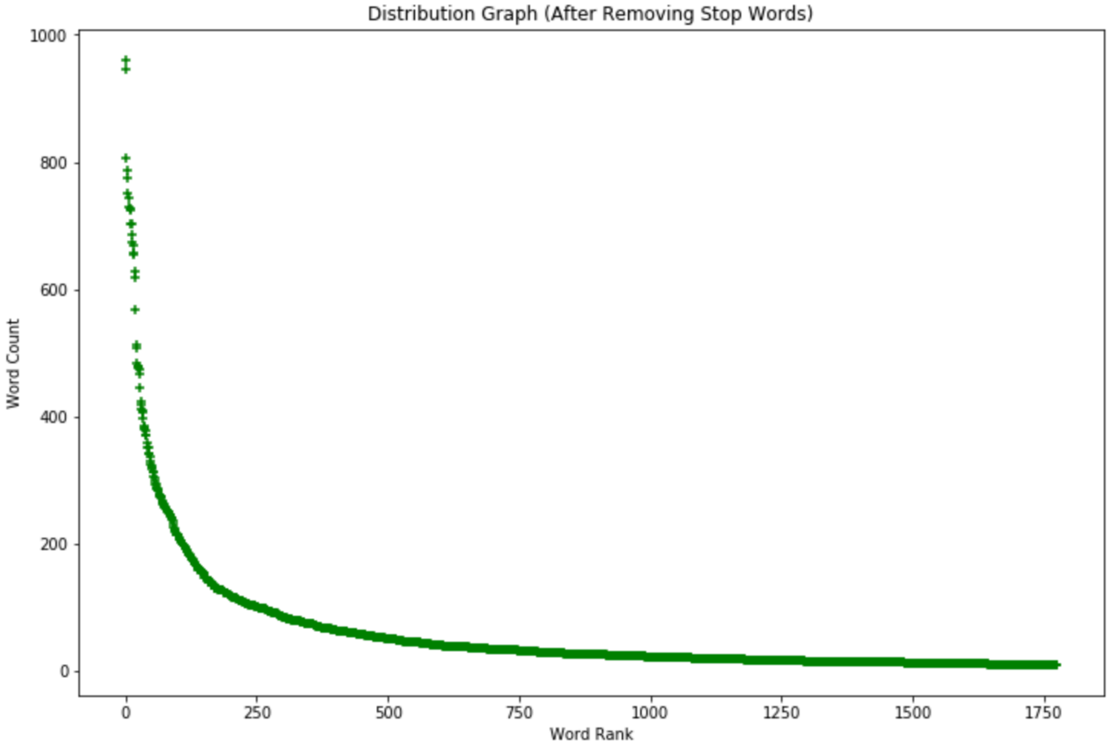


2 Identify Stop Words, Document Frequency Threshold

Stop Words	`stop_words = ['this','for','the','has','had','have','is','was','i','am','a','to','an','are','me','by','my','him',`
	`'on','in','them','where','you','be','can','at','there','here','if','or','they','out','from',`
	`'where','of','we','were','all','as','and','not','but','so','when','with','until','that','he','she',`
	`'it','what','has','your','am','why','our']`

Max Document Frequency	**max_df=0.5** Any word which is repeated over 50% (=2000 * 50% = **1000 of Document**) will be ignored, after the stop words
Min Document Frequency	**min_df=10** Any word which is repeated below **10 of Document** will be ignored, after the stop words

3 Distribution Graph (Words Count Vs Word Rank) – After Stop Words + applied max_df & min_df



4 Code Snippet

```python
def stop_words_create():
    stop_words = ['this','for','the','has','had','have','is','was','i','am','a','to','an','are','me','by','my','him',
                  'on','in','them','where','you','be','can','at','there','here','if','or','they','out','from',
                  'where','of','we','were','all','as','and','not','but','so','when','with','until','that','he','she',
                  'it','what','has','your','am','why','our']
    return stop_words


def data_vectorize(X,stop_words,max_df=1.0,min_df=1,train_idx=2000):
    X_lower = list(np.char.lower(X.astype(str)))
    vectorizer = CountVectorizer(stop_words=stop_words,max_df=max_df,min_df=min_df)
    data_feature=vectorizer.fit_transform(X_lower).toarray()
    words_freq = np.sum(data_feature[:train_idx],axis=0)
    words = np.array(vectorizer.get_feature_names())
    df_words_dist = pd.DataFrame(words,columns=['words'])
    df_words_dist['words_freq'] = words_freq
    #word_freq=-np.sort(-np.sum(data_feature_arr,axis=0))
    return df_words_dist.sort_values(by='words_freq',ascending=False),data_feature[:train_idx].astype(int),data_feature[train_idx:].astype(int)


stop_words = stop_words_create()
df_words_dist_after_sw,X_features_train_after_sw,X_features_test_after_sw=data_vectorize(X,stop_words,max_df=0.5,min_df=10,train_idx=2000)


def cosine_distance(X_features_train,X_features_test,metric='cosine',n_neighbors=5):
    neigh=NearestNeighbors(metric='cosine',n_neighbors=n_neighbors)
    neigh.fit(X_features_train)
    cosine_score,cosine_index=neigh.kneighbors(X_features_test)
    return cosine_score,cosine_index
```

5 Review with Score

Below are the score against the 5 documents with lower cosine distance (Close to the document) against the document “Horrible customer service”

Original Review	Distance Score
[['Rogers ...\n\n1) is over priced\n2) have horrible customer service\n3) faulty and incorrect billing\n4) poor customer service\n5) not enough options\n6) never arrive for an appointment'	0.4444444444444443
['Horrible service, horrible customer service, and horrible quality of service! Do not waste your time or money using this company for your pool needs. Dan (602)363-8267 broke my pool filtration system and left it in a nonworking condition. He will not repair the issue he caused, and told me to go somewhere else. \n\nSave yourself the hassle, there are plenty of other quality pool companies out there. \n\nTake care!'	0.611111111111111
["Service was horrible came with a major attitude. Payed 30 for lasagna and was no where worth it. Won't ever be going back and will NEVER recommend this place. was treated absolutely horrible. Horrible."	0.6631392315733924
['Went to Marca today to get a haircut and was given a great service both by front desk - customer service and by Georgia, girl who did my hair. I guess I got lucky with her as she has years of experience doing this job. She has excellent customer service skills and takes excellent care of her customers.'	0.6792498504502079
["The service is horrible. It's not bad inside, but really one of the most annoying clubs in Vegas. I'm all for Vegas clubs, but service here sucks."	0.7113248654051871

6 Query Result

Below are the documents which are selected against the document “Horrible customer service”. Reason for being chosen, the cosine distance is low, means these statements are very close the statement of “Horrible customer service”. Also as a double check, the star rating also given as “1” which seems to have bad experience / bad service.

Original Review	Star	Distance Score
[['Rogers ...\n\n1) is over priced\n2) have horrible customer service\n3) faulty and incorrect billing\n4) poor customer service\n5) not enough options\n6) never arrive for an appointment'	**1**	0.4444444444444443
['Horrible service, horrible customer service, and horrible quality of service! Do not waste your time or money using this company for your pool needs. Dan (602)363-8267 broke my pool filtration system and left it in a nonworking condition. He will not repair the issue he caused, and told me to go somewhere else. \n\nSave yourself the hassle, there are plenty of other quality pool companies out there. \n\nTake care!'	**1**	0.611111111111111
["Service was horrible came with a major attitude. Payed 30 for lasagna and was no where worth it. Won't ever be going back and will NEVER recommend this place. was treated absolutely horrible. Horrible."	**1**	0.6631392315733924
["The service is horrible. It's not bad inside, but really one of the most annoying clubs in Vegas. I'm all for Vegas clubs, but service here sucks."	**1**	0.7113248654051871

Below sentence is not selected, because it’s a good feedback, however since the word “customer” and “service” present, the cosine distance is low and seems closure to the word “Horrible customer service”. Also looking into other dimension, the star rating is shows as “5”, which seems very satisfaction rating.

Original Review	Star	Distance Score
['Went to Marca today to get a haircut and was given a great service both by front desk - customer service and by Georgia, girl who did my hair. I guess I got lucky with her as she has years of experience doing this job. She has excellent customer service skills and takes excellent care of her customers.'	**5**	0.6792498504502079

7 Accuracy with threshold 0.5

```
def traintest_split(df_X,df_y,split_ratio=0.1):
    df_X_train,df_X_test,df_y_train,df_y_test=train_test_split(df_X,d
f_y,test_size=split_ratio,random_state=2019)
    return df_X_train,df_X_test,df_y_train,df_y_test


def logistic_model(df_X_train,df_y_train):
    clf=LogisticRegression(random_state=2019)
    clf.fit(df_X_train,df_y_train)
    return clf


def logistic_pred(clf,df_X_train,df_y_train,df_X_test,df_y_test,thres
hold,print_ind=False):
    clf_pred_train = []
    clf_pred_test = []

    clf_prob_train=clf.predict_proba(df_X_train)[:,0]
    clf_prob_test=clf.predict_proba(df_X_test)[:,0]

    #Predict for Train Accuracy
    for i in range(clf_prob_train.shape[0]):
        if (clf_prob_train[i] > threshold):
            clf_pred_train.append(1)
        else:
            clf_pred_train.append(5)

    #Predict for Test Accuracy
    for i in range(clf_prob_test.shape[0]):
        if (clf_prob_test[i] > threshold):
            clf_pred_test.append(1)
        else:
            clf_pred_test.append(5)

    train_acc = (np.sum(clf_pred_train == df_y_train)/df_y_train.shap
e[0])*100
    test_acc = (np.sum(clf_pred_test == df_y_test)/df_y_test.shape[0]
)*100
    return train_acc,test_acc,clf_pred_train,clf_pred_test,clf_prob_t
rain,clf_prob_test
```

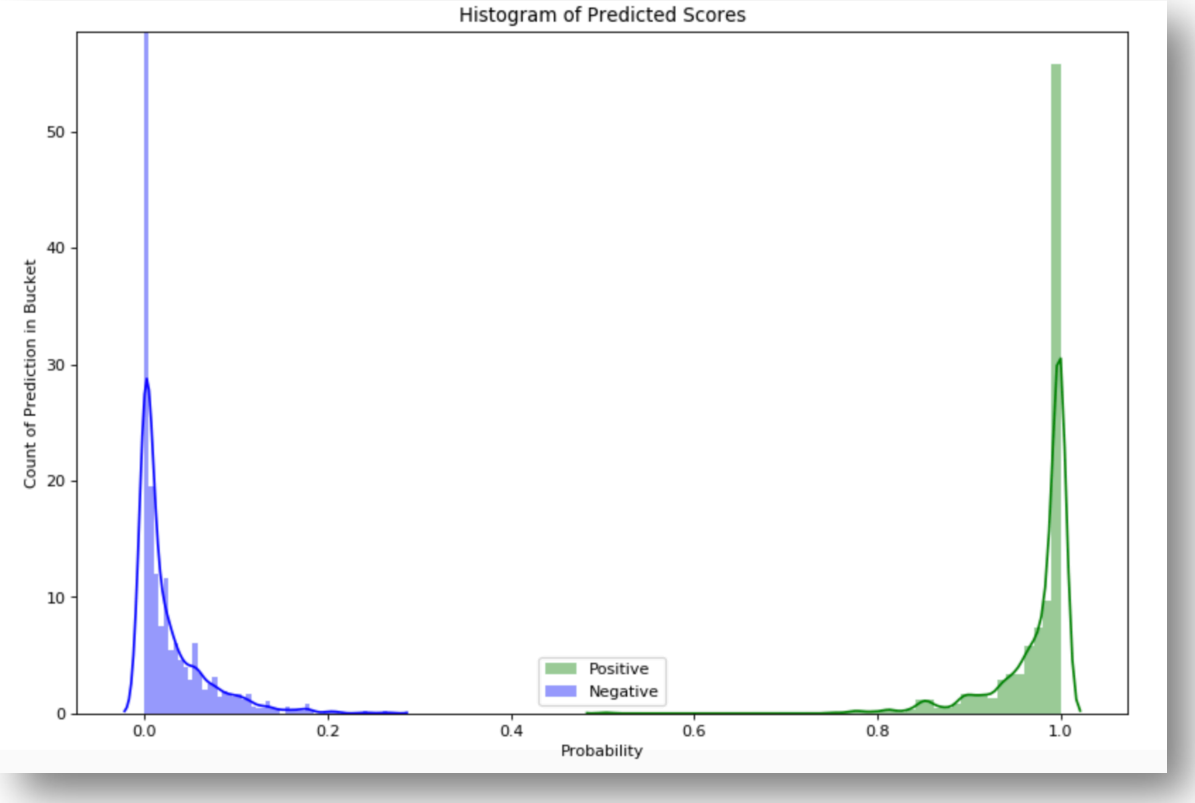
```
df_X_features=pd.DataFrame(X_features)
df_y=pd.DataFrame(y)
df_X_train,df_X_test,df_y_train,df_y_test=traintest_split(df_X_featur
es,df_y)
clf=logistic_model(df_X_train,df_y_train)
train_acc,test_acc,clf_train_pred,clf_test_pred,clf_prob_train,clf_pr
ob_test=logistic_pred(clf,df_X_train,df_y_train,df_X_test,df_y_test,t
hreshold=0.5)
df_X_train['pred']=clf_train_pred
df_X_test['pred']=clf_test_pred
df_final=pd.concat([df_X_train,df_X_test])
print ("Train Accuracy: {}".format(train_acc[0]))
print ("Test Accuracy: {}".format(test_acc[0]))

Train Accuracy: 100.0
Test Accuracy: 95.0
```

8 Predicted Scores

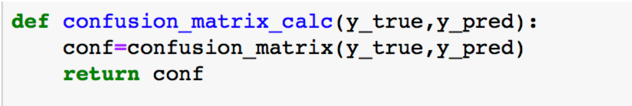
```
def distplot_clf(clf_prob_train,clf_prob_test,threshold):
    plt.figure(figsize=(12,8),dpi=80)
    sns.distplot(clf_prob_train[clf_prob_train >= threshold],bins=50, color='green',label='Positive')
    sns.distplot(clf_prob_train[clf_prob_train < threshold],bins=50,color='blue',label='Negative')
    plt.xlabel('Probability')
    plt.ylabel('Count of Prediction in Bucket')
    plt.title('Histogram of Predicted Scores')
    plt.legend()
```

```
distplot_clf(clf_prob_train,clf_prob_test,threshold=0.5)
```

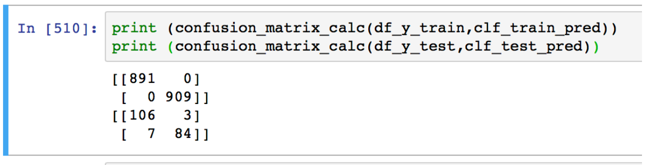


9 Accuracy again and curve

Different Threshold	**0.47**11830818835384
Train Accuracy	**100**
Test Accuracy	**95.0**
Why Chosen this Threshold	The new threshold is calculated from the Youden's J statsitic, after deriving the FPR(“False Positive Rate) and TPR (“True Positive Rate”) and thresholds. The Youden’s J statistics is the point on the ROC Curve (with value of FPR and TPR), which have maximum distance from line of equality (i.e. from the diagonal line). The distance from the diagonal till the ROC curve is calculated as = np.argmax(tpr – fpr), where the index comes as 15 and the value of the thresholds @ index 15 comes as **0.4711830818835384**. Also after checking the number of False Positive and True Positive in the **Test Sets (Out of 200 Samples)**, it’s observed that there are only **3** True Negative which misclassified as Positive (False Positive) and **84** are correctly classified as Positive (out of 91)

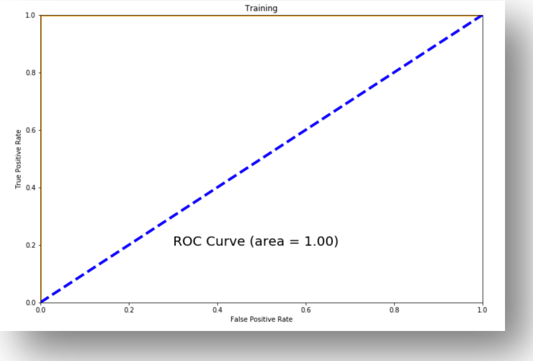
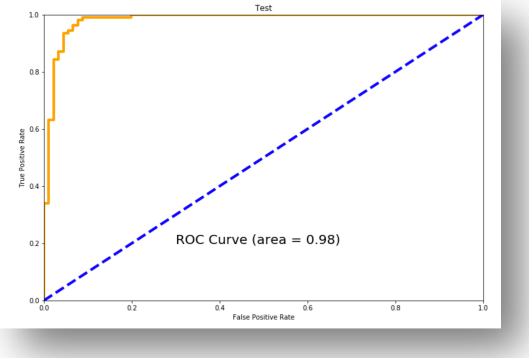


```
def confusion_matrix_calc(y_true,y_pred):
    conf=confusion_matrix(y_true,y_pred)
    return conf
```



```
In [510]: print (confusion_matrix_calc(df_y_train,clf_train_pred))
          print (confusion_matrix_calc(df_y_test,clf_test_pred))

          [[891   0]
           [  0 909]]
          [[106   3]
           [  7  84]]
```

ROC AUC Curve (Training)	ROC AUC Curve (Test)
	

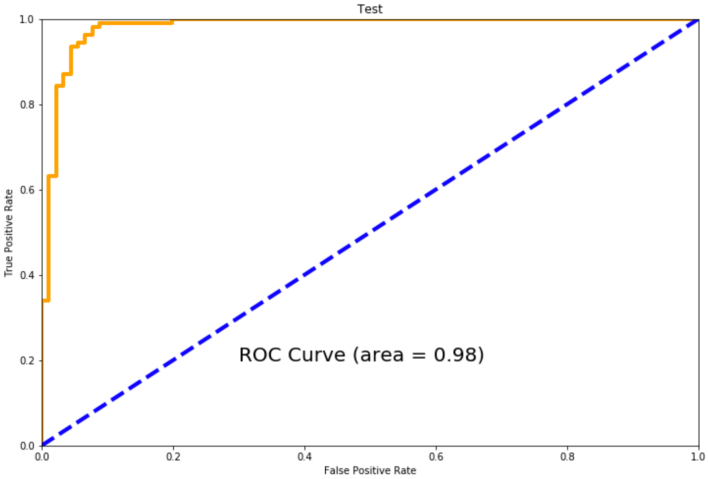
10 Best Threshold

The Best Thresholds = **0.4711830818835384**

Reason for being Best Thresholds: Below are the reasons why the thresholds is chosen as best thresholds

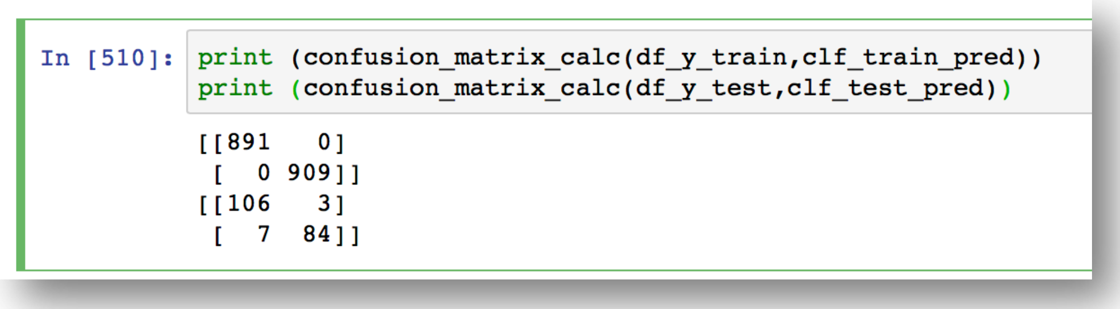
1. The distance between the ROC curve and line of equality (diagonal line) has the maximum distance, which means the point is far from the line of equality and may have less False Positive Rate (FPR) and more True Positive Rate

2. AUC is **0.98**, higher the value more accurate prediction



3. Number of misclassified Positive (False Positive) is only **3**

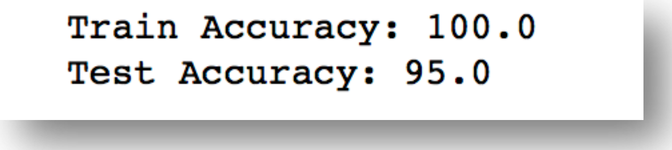
4. Number of correct classified Positive (True Positive) is **84** out of 91 (84/91)



```
In [510]: print (confusion_matrix_calc(df_y_train,clf_train_pred))
          print (confusion_matrix_calc(df_y_test,clf_test_pred))

          [[891   0]
           [  0 909]]
          [[106   3]
           [  7  84]]
```

5. Train Accuracy is **100** , and Test Accuracy is **95.0**



```
Train Accuracy: 100.0
Test Accuracy: 95.0
```