International Journal of Technology, Science and Engineering **ISSN: 2457-1016**

Vol. 3, Issue I, Jan-Mar, 2020 http://www.bharatpublication.com/journal-detail.php?jID=25/IJTSE

LEVERAGING THE MACHINE LEARNING TOOLS AND TECHNIQUES IN THE EFFICACIOUS PREDICTION OF TESTS MESSAGE SEQUENCE

Abhishek Dhillon

G D Goenka Public School, Sector-22, Rohini, New Delhi, India

ABSTRACT

Numerous applications exist for predicting the subsequent message following a sequence of messages. Natural language processing (NLP), log anomaly detection, automated customer service, and numerous other fields can all benefit from its use. Computerizing Client. The administration is one such application which will be in an extraordinary concentration in this paper. Automating customer service can save a lot of money and time for any business. It makes it take longer for customers or users to get a response, which makes them happier. It lightens the company's load. In this paper, we automate the subsequent debug message or step required to resolve a customer issue or ticket using a machine learning algorithm (Support vector machine + gridsearchcv). Additionally, this algorithm can predict multiple subsequent steps, automating the customer service team's debugging efforts. Given a sequence, it accurately predicts the subsequent debug message at 98.08 per cent.

INTRODUCTION

Customer service automation automatically resolves user issues without having to speak with a human being. Every company's customer service department answers a lot of questions every day. The majority of those requests are common and frequently answered. The customer service team can only respond to more complex questions that require human intervention if it can automate repetitive queries. The customer service team can handle problems that require specialized attention, which may be small. Businesses can save money by using a small customer service team. Giving customers or users debug messages or steps to solve the problem is usually how a customer service team operates. A method for creating a model that can predict the next step or debug message is described in this paper. We used the information technology incident management process's event log to train the model. The first sequence of debug messages serves as the model's initial input. After that can use it multiple times to generate the subsequent debug messages or steps necessary to resolve the customer issue.

DATA

A. Data Structure Data is an event log of the IT incident management procedure [4]. It includes the various debug messages and steps to close the customer ticket. You must complete anywhere from one to 78 debug messages or steps to close a customer ticket. The data are in CSV format. There are five columns and approximately 4.6 million rows. Ticket Number, Status, Time, Group Name, and Owner are the five columns. The status will represent the most crucial data for the model's development. The sample dataset is shown in Figure 1.

	A	B	C	D	E
1	TicketNum	Status	Time	GroupName	Owner
2	5078917	Open	7/1/2010 0:00	GRP01	RES01
3	5078917	acknowledged notification	7/1/2010 5:18	GRP01	RES11
4	5078917	Assignment	7/1/2010 5:21	GRP01	RES21
5	5078917	analysis/research/tech note	7/1/2010 5:21	GRP01	RES21
6	5078917	Assignment	7/1/2010 16:37	GRP01	RES31
7	5078917	pending customer	7/1/2010 16:37	GRP01	RES31
8	5078917	Assignment	7/6/2010 16:40	GRP01	RES31
9	5078917	reassigned-misrouted	7/6/2010 16:40	GRP01	RES31
10	5078917	reassigned-misrouted	7/6/2010 16:40	GRP01	RES31
11	5078917	acknowledged notification	7/6/2010 16:42	GRP01	RES31
12	5078917	Assignment	7/6/2010 16:54	GRP01	RES31
13	5078917	analysis/research/tech note	7/6/2010 16:54	GRP01	RES31
14	5078917	restored to service	7/6/2010 17:54	GRP01	RES31
15	5078917	Closed	7/6/2010 18:00	GRP01	RES31
16	5078921	Open	7/1/2010 0:01	GRP11	RES41
17	5078921	acknowledged notification	7/1/2010 0:02	GRP11	RES51
18	5078921	analysis/research/tech note	7/1/2010 0:02	GRP11	RES51
19	5078921	Closed	7/1/2010 0:25	GRP11	RES51
20	5078922	Open	7/1/2010 0:01	GRP11	RES41
21	5078922	acknowledged notification	7/1/2010 0:02	GRP11	RES51
22	5078922	analysis/research/tech note	7/1/2010 0:02	GRP11	RES51
23	5078922	Closed	7/1/2010 9:42	GRP11	RES51
24	5078971	Open	7/1/2010 1:09	GRP21	RES61
25	5078971	Assignment	7/1/2010 1:10	GRP21	RES11
26	5078971	analysis/research/tech note	7/1/2010 1:10	GRP21	RES11
27	5078971	alert stage 1	7/1/2010 1:30	GRP21	RES11

Fig 1: Data Sample

Pre-processing the Data

The data are changed to fit the model. The data is first separated into sequences and embedded in the dictionary during this procedure. The code necessary to complete the preceding step is depicted in figure 2. When the word reference has been populated with information, the subsequent stage transforms the string information into a mathematical structure. To accomplish this, we assign a numerical value to the different debug messages and steps derived from the dataset. Figure 3 shows the remarkable troubleshooting messages/steps that have been utilized in addressing client tickets.

```
for row in csvreader:


    if row[1]=="Status":
        continue;

    listofkeys.append(row[1])
    if row[1]=="Open":
        i=i+1
        templist = []
    elif row[1] == "Closed":

        datasequence[i] = templist
        #print(datasequence)
        #print(templist)


    else:
        templist.append(row[1])
```

Figure 2

```
['', 'pending release', 'communication with customer', 'Open', 'pending repair', 'pending confirmation', 'Work in Progress', 'analysis/research/tech note', 'pe
nding vendor', 'Manually Acknowledged', 'Acknowledged', 'Pending Vendor', 'restarted notification', 'communication with provider', 'alert stage 1', 'Assignment
', 'Closed', 'automated', 'acknowledged notification', 'reassigned-misrouted', 'alert stage 2', 'Restored to Service', 'equipment return', 'Pending Customer',
'reassigned-addl work required', 'communication with vendor', 'waiting parts', 'restored to service', 'pending provider', 'pending customer', 'status request',
 'DEADLINE ALERT', 'alert stage 3']
```

Figure 3

IMPLEMENTATION

A. Data Scaling Data scaling is a crucial step that needs to be taken before we train the model. Scaling is the process of transforming data into a standard format. Consider an instance in which two currencies are involved: the yuan and the USD.

```
st_x= StandardScaler()
x_train= st_x.fit_transform(X_train)
x_test= st_x.transform(X_test)
parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}
svc = svm.SVC()
clf = GridSearchCV(svc, parameters).fit(x_train,Y_train)
```

Figure 4

For effective mathematical operations, it is essential that these two currencies be converted to the same unit—in this case, the yuan or the US dollar—when the operations are carried out on them. 1 US dollar equals 7 yuan, for example. The code snippet for scaling data is depicted in Figure 4.

1.) Keying in the distinctive debug messages and steps: The unique debug messages and steps are given unique keys, as depicted in figure 3. The classes that we will try to predict are these keys. This is a multiclass classifier problem because, as shown, there are multiple keys and classes.

2.) SVM or Support Vector Machine: It's a supervised machine learning algorithm that can solve binary and multiclass classification problems. It is one of the heartiest expectation strategies which utilizes the VC hypothesis (factual learning structure)

for arranging.

3.) CV Grid Search: The engineer's choice of hyperparameters determines how well any model performs. Remembering that we must choose a model's hyperparameters in advance is also important. The GridSearchCV algorithm is used to determine an algorithm's optimal hyperparameters. Figure 5 depicts the hyperparameters for the support vector machine algorithm mentioned earlier [5]. Cross-validating the model, GridSearchCV determines the best hyperparameter combinations and evaluates the model. Figure 6 shows the code scrap of fitting the information utilizing gridsearchCV.

```
{ 'C': [0.1, 1, 10, 100, 1000],
  'gamma': [1, 0.1, 0.01, 0.001, 0.0001],
  'kernel': ['rbf','linear','sigmoid'] }
```

Figure 5

```
svc = svm.SVC()
clf = GridSearchCV(svc, parameters).fit(x_train,Y_train)
```

Figure 6

RESULTS

The f1-score, precision, and recall scores are used to assess a model's robustness. The F1-measure provides a numerical number that considers both precision and recalls score [6]. The precision score represents the number of positive class predictions that belong to the positive class. The total number of correct positive class predictions is represented or quantified by the recall. Figure 7 depicts how the confusion matrix organizes all these data in tabular form.

A machine learning model's accuracy score is one more important factor that is taken into consideration when evaluating its effectiveness. The total number of correct predictions divided by the total number of data points is used to calculate the accuracy. Figure 7 depicts the model's accuracy of 98.08 per cent.

accuracy is 0.9808823529411764

	precision	recall	f1-score	support
2	1.00	0.91	0.95	43
3	1.00	0.97	0.99	34
5	1.00	1.00	1.00	14
6	1.00	1.00	1.00	10
8	1.00	0.92	0.96	51
9	1.00	1.00	1.00	1
13	1.00	1.00	1.00	2
15	1.00	1.00	1.00	1
19	1.00	1.00	1.00	1
25	1.00	1.00	1.00	1
28	0.98	0.98	0.98	219
30	0.97	1.00	0.99	303
accuracy			0.98	680
macro avg	1.00	0.98	0.99	680
weighted avg	0.98	0.98	0.98	680

Figure 7

CONCLUSIONS

For any customer ticket, anticipating the subsequent debug messages and steps has significant business benefits. The trained model predicts the subsequent debug message with a high accuracy of 98.08 per cent in this paper. The model has been trained on over 54000 cases or issues altogether. The customer service team has provided customers over 55 debug messages and steps to close the ticket for each case or issue. Numerous issues have arisen throughout the model's extensive training. Can run this model into a circle until it close is anticipated and utilized as a complete robotizing device for creating the troubleshooting messages. As a result, we can utilize the above support vector machine method in conjunction with gridsearchcv to automate repetitive issues that have already been resolved.

REFERENCES

[1] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 1285–1298. https://doi.org/10.1145/3133956.3134015.

[2] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning," Advances in neural information processing systems 28 (2015).

[3] Q. Fu, J. -G. Lou, Y. Wang and J. Li, "Execution Anomaly Detection in Distributed Systems through Unstructured Log Analysis," 2009 Ninth IEEE International Conference on Data Mining, 2009, pp. 149-158, doi: 10.1109/ICDM.2009.60.

[4] Event log of IT Incident management process. [Online]. Available: https://www.kaggle.com/datasets/asjad99/it-incident-management-process

[5] Hyperparameters for support vector machine (SVM). [Online]. Available: https://www.mygreatlearning.com/blog/gridsearchcv

[6] Definition of precision score, f1 score and recall score. [Online]. Available: https://machinelearningmastery.com/precision-recall-and-f-measure-forimbalanced-classification.