

Classification of Student Query using Machine Learning

Voore Saithanish¹, K. Sai Varun², Dr. M. Senthil Kumaran³

¹⁻²Student, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, Tamil Nadu, India

³Professor, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, Tamil Nadu, India

Abstract – The Educational institutions and universities were getting bulk amount of data in the form of queries send by students regarding their academics and educational issues. Because of this huge data it is difficult for the universities to classify, sort and resolve which takes much amount of time. This Project algorithm which works for classifying the data into their respective departments using Machine Learning Algorithm in the way assigning Keywords for the data then sorting them into the category. So, the students get resolved their queries in short span of time by classifying their quires directly to their respective Departments.

Key Words: Classification, Text Processing, Machine Learning, TF-IDF (term frequency-inverse document frequency), Data Analysis, SVM (support vector machine)

1. INTRODUCTION

The data received from students to the universities in daily bias in the bulk form which makes the universities difficult to sort out the queries according the departments, taking huge amount of time and complexity in classifying the data.

The data in the fields of students queries in every department as the fee issues, transportation, library and many more in this form. This type of data is much complex to find out and resolve in a period of time. The students facing problems as well as the time period of resolving their queries is delay too. So, by this project where it is designed to classify the data into the departments by giving the data keywords and making into the sub groups which the algorithm differentiates the data into types of departments that makes them easier to sort them out. The query raised by the students is stored in a database where it is received from a website, having the terms as student name, class, reg no, department, mail, category, and the complaint data, priority.

The data given by the student is then received by the category department with the priority and the students receives the notification of his/her status of the query. The department gets informed regarding the query, time posted, priority which makes the department easier to resolve the query. After the query resolved the status of the query is seen by the student whether it is solved, in progress, hold, etc.

The TF-IDF (term frequency-inverse document frequency) classification algorithm is used to classify the data into the category using the label number and names given in form of vectors which are converted from the data form by the algorithm. This makes the task easier and faster in finding the query related to the category that makes the students issues resolve in time and making the task simple for the management.

1.1 Objective

The main objective is to make the task easy and in short span time and in the way helping both the students and management as

- Students get their queries resolved in short time and,
- Managements find it easy to classify the data and resolving them.
- Using the Machine learning and cutting-edge technologies in daily life situations and making them easier and faster.

2 Problem Statement

In every educational institution, there will be Many queries for students regarding the technical or administration and other categories. So, to clear the student query in a quick and easy manner this algorithm helps the institution to classify the student posted queries to respective departments.

The time delay in resolving the problems is no more and the process is in a lucid way. No more confusions and complex situations as clashing the queries and not able to find one in a bulk file.

3. Algorithm

Input:

D: grumblings information (comprises of the relative multitude of grievances)

Yield:

Weight Matrix (which comprises of the multitude of loads of terms are

called vectors)

Method:

1-for every grumbling archive (ci) do

2-for each term (tj) in ci do

3-TF-IDF score for term tj

in record

ci = TF (ci

, tj) * IDF (tj)

Where, IDF = Inverse Document Frequency

TF = Term Frequency

TF (ci

, tj) = (Term tj recurrence in record ci)

(Complete words in archive ci)

$IDF (ci) = \log_2 ((Total Documents) / (records$

With term tj))

4-End for of term

5-End for of objection record

6-The vectors are put away in an exhibit for preparing and testing

purposes, during arrangement.

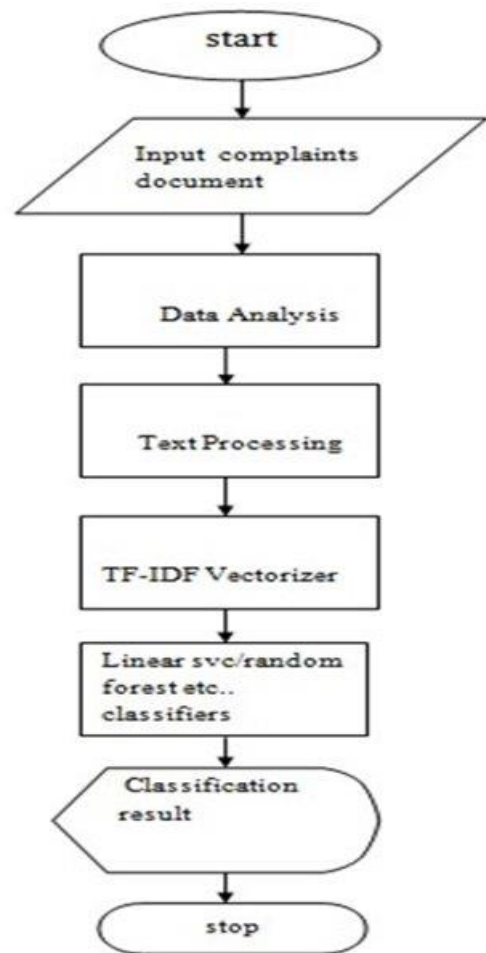


Chart -1: Flow Chart

4. Project Description

The complaints are in text format; in order to classify them using a classification method, the text must be translated into vectors. to be able to foresee the class We use TF-IDF to accomplish this. TF-IDF is a method for converting text to vectors. The inverse document frequency is used to find the frequency of a document. Determine which terms are the most relevant to a particular issue. It's a unique situation. Statistics are used to determine how relevant a term or word is. refers to a document in a corpus or a collection of documents. The TF-IDF of a word in a document is determined using two indices IDF (inverse term frequency) and TF (term frequency) The term frequency (TF) is calculated by counting the number of times a word appears in a document and adjusting the frequency for the document's length or number of words. IDF (inverse document frequency) of a word or phrase the term denotes how uncommon or uncommon a word is throughout the entire dictionary. A corpus is a group of documents. This can be computed by dividing the number of papers by the total number of documents. The word occurred in a significant number of documents. If a word or term appears in a large number of places in the

manuscript, it's a good sign. If it's highly common, it's scaled to '0,' else it's scaled to '1.' We can get the result by multiplying the two terms together.

The TD-IDF score the greater the score, the more relevant it is. After translating the text, we use techniques such as Random Forest Classifier, Linear SVC, Multinomial NB, and Logistic Regression to classify it. Regression The "complaints.csv" data collection will contain the Token No., Date, Year, Student-ID, Email Id, and other attributes Category of Complaints Cat, Issue Resolver, Counselor Name, Issue Date, and Issue, Number of Days to Resolve Status: Completed, Status: Completed, Status: Completed, Status: Completed, Status: Completed, Using the "complaints.csv" file dataset, we'll create a new Data Frame with the following elements:

(Categories includes health issues, the examination part, and so on detention, etc.) and the Grievance Category, which includes a comprehensive grievance Now we'll get rid of the duplicates in the database.

```
In [28]: mean_accuracy = cv_df.groupby('model_name').accuracy.mean()
std_accuracy = cv_df.groupby('model_name').accuracy.std()

acc = pd.concat([mean_accuracy, std_accuracy], axis=1,
                ignore_index=True)
acc.columns = ['Mean Accuracy', 'Standard deviation']
acc

Out[28]:
```

model_name	Mean Accuracy	Standard deviation
LinearSVC	0.865829	0.031106
LogisticRegression	0.851429	0.014636
MultinomialNB	0.843429	0.014791
RandomForestClassifier	0.616533	0.047004

```
In [29]: plt.figure(figsize=(8,5))
sns.boxplot(x='model_name', y='accuracy',
            data=cv_df,
            color='lightblue',
            showmeans=True)
plt.title("MEAN ACCURACY (cv = 5)\n", size=14);
```

Fig -01: The accuracy and deviation shown as output

Assign a unique Id to the newly formed Data Frame, let's call it "df1." making a temporary for each category in other works a dictionary for future use. We can now see which section or department is receiving the most complaints from students. Now we'll put the theory into practice. TFID-Vectorizer, which converts each complaint into a vector. We'll store the vectors in an array, and we'll use them later. can find out how many Unigrams and Bigrams there are. Following that, we will create a map. the Unigrams and Bigrams with the most connected

Remove the stop words from each complaint. The division of Data for Training and Testing will be collected in the same way as 'X' is collected. Having all of the Grievance Categories, as well as 'y', which is made up of We need to

forecast the labels of the target labels. Everything is completed at this point. will be sorted out by data training and assessment Now we use a variety of machine learning classification methods to forecast the outcome of the complaints. The other is now. Maintaining the database for sending the messages is an element of the project. Regarding the complaints, bidirectional notification is required. As a result, When the categorization procedure is finished, the anticipated results are displayed. We'll take the output and make a prediction based on it. cause a notification to be sent to that department's employee who will be responsible for resolving the issue Finally, once the complaint has been resolved, resolved, and the issue has been posted on the website The issue raiser will be notified, and work will begin. will be performed quickly and without wasting time, and when compared to other complaint classifiers, it will be the best. interaction between two people on a one-to-one basis.

```
In [24]: df1
Out[24]:
```

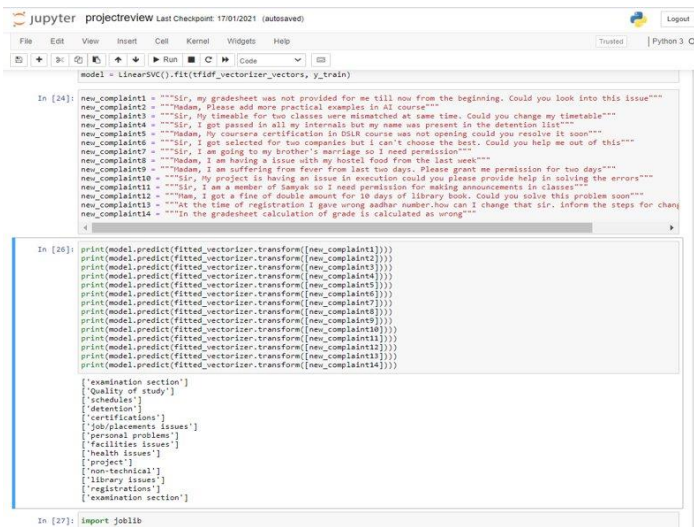
	cat	Grievance Category	category_id
0	examination section	hello sir I have done with my fee payment but ...	0
1	health issues	sir,I have full headache and fever,I am feelin...	1
2	examination section	In Gradesheet with respective to department hi...	0
3	health issues	i have been absent for last two weeks due to m...	1
4	Quality of study	sir I am from s13 section sir our oops faculty...	2
...
621	personal problems	sir I have brother marrage next week and I was...	8
622	personal problems	sir I am suffering with full fever and headach...	8
623	personal problems	sir since the last 3 days I am having the symp...	8
624	personal problems	I am going for applying the education loan so ...	8
625	personal problems	sir I have brother marrage next week and I was...	8

626 rows x 3 columns

Fig -02: The output shows the sorting of data as of category_id

5. Result

As the queries received from the students, they were analyzed and classified to the departments mentioned according to the query which were converted to vectors to identify the category then were classified and shown as in the figure below the departments were shown.



```

model = LinearSVC().fit(tfidf_vectorizer_vectors, y_train)

In [24]: new_complaint1 = ""Sir, my gradesheet was not provided for me till now from the beginning. Could you look into this issue""
new_complaint2 = ""Madam, please add more practical examples in AI course""
new_complaint3 = ""Sir, My timetable for two classes were mismatched at same time. Could you change my timetable""
new_complaint4 = ""Sir, I got passed in all my internals but my name was present in the detention list""
new_complaint5 = ""Madam, My course certification in ODR course was not opening could you resolve it soon""
new_complaint6 = ""Sir, I got selected for two companies but I can't choose the best. Could you help me out of this""
new_complaint7 = ""Sir, I am going to my brother's marriage so I need permission""
new_complaint8 = ""Madam, I am having an issue with my hostel food from the last week""
new_complaint9 = ""Madam, I am suffering from fever from last two days. Please grant me permission for two days""
new_complaint10 = ""Sir, My project is having an issue in execution could you please provide help in solving the errors""
new_complaint11 = ""Sir, I am a member of Sanyak so I need permission for making announcements in classes""
new_complaint12 = ""New, I got a fine of double amount for 10 days of library book. Could you solve this problem soon""
new_complaint13 = ""At the time of registration I gave wrong aadhar number.how can I change that sir. inform the steps for chang
new_complaint14 = ""In the gradesheet calculation of grade is calculated as wrong""

In [26]: print(model.predict(fitted_vectorizer.transform([new_complaint1])))
print(model.predict(fitted_vectorizer.transform([new_complaint2])))
print(model.predict(fitted_vectorizer.transform([new_complaint3])))
print(model.predict(fitted_vectorizer.transform([new_complaint4])))
print(model.predict(fitted_vectorizer.transform([new_complaint5])))
print(model.predict(fitted_vectorizer.transform([new_complaint6])))
print(model.predict(fitted_vectorizer.transform([new_complaint7])))
print(model.predict(fitted_vectorizer.transform([new_complaint8])))
print(model.predict(fitted_vectorizer.transform([new_complaint9])))
print(model.predict(fitted_vectorizer.transform([new_complaint10])))
print(model.predict(fitted_vectorizer.transform([new_complaint11])))
print(model.predict(fitted_vectorizer.transform([new_complaint12])))
print(model.predict(fitted_vectorizer.transform([new_complaint13])))
print(model.predict(fitted_vectorizer.transform([new_complaint14])))

[examination section]
[Quality of study]
[schedules]
[detention]
[certifications]
[job/placements issues]
[personal problems]
[facilities issues]
[health issues]
[project]
[non-technical]
[library issues]
[registrations]
[examination section]

In [27]: !import joblib
    
```

Fig -03: The Queries Classified into Respective Departments

6. Conclusion

The student query classification system using Linear SVC with the combination of TF-IDF (Term Frequency-Inverse Document Frequency) as results in giving the classification of data in the database according to the category which were divided by the use of vector notation assigned for the data that makes sorting the data easier. The interface jupyter notebook is used to read and take the data and giving the output in the forms of tables and graphs for the respective queries. Using machine learning we make the query collection and classification simple and this is widely used technology now-a-days. This model results in accuracy of 89% and efficient in working the data in the bulk form. This helps in reducing the time factor and for the benefit of students and organizations both.

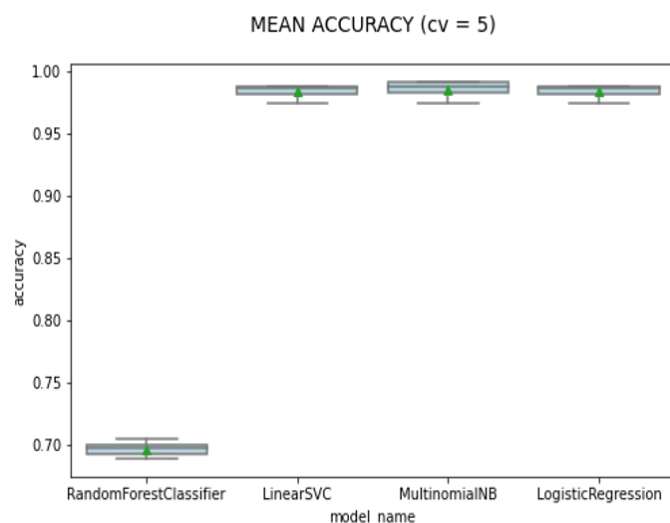


Fig -1: Accuracy Graph

References

1. N. S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3):175-185.
2. Koray Balci -Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey Albert Ali Salah - Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey Automatic Classification of Player Complaints in Social Games.
3. Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.
4. M.A. Fauzi, Automatic complaint classification system using classifier ensembles, January 2018.
5. Ganesan, Kavita, and Guangyu Zhou. (2016), "Linguistic Understanding of Complaints and Praises in User Reviews.", Proceedings of NAACLHLT.
6. Imam Cholissodin, Maya Kurniawati, Indriati, Issa Arwani Informatics Department, PTIIK, Brawijaya University, Malang, Indonesia. Classification of Campus E-Complaint Documents using Directed Acyclic Graph Multi-Class SVM Based on Analytic Hierarchy Process 2014.
7. Moschitti, A., & Basili, R. (2004), "Complex Linguistic Features for Text Classification: A Comprehensive Study.", Advances in Information Retrieval, 181-19.
8. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). "Deep Learning for Hate Speech Detection in Tweets", Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17.
9. Ryan M. Eshleman and Hui Yang. 2014 "Hey #311, Come Clean My Street! ": A Spatio-temporal Sentiment Analysis of Twitter Data and 311 Civil Complaints. In 2014 IEEE Fourth International Conference on Big Data and Cloud Computing, pages 477- 484.
10. Ahmad Fauzan and Masayu Leylia Khodra. 2014. Automatic Multilabel Categorization using Learning to Rank Framework for Complaint Text on Bandung Government. In 2014 Int. Conf. of Advanced Informatics: Concept, Theory and Application (ICAICTA), pages 28-33. Institut Teknologi Bandung, IEEE.
11. Ana Catarina Forte and Pavel B. Brazdil. 2016. Determining the Level of Clients' Dissatisfaction from Their Commentaries. In Computational Processing of the Portuguese Language - 12th Int. Conf., PROPOR 2016, volume 9727 of Lecture Notes in Computer Science, pages

74–85. Springer. (Basic Book/Monograph Online Sources)
J. K. Author. (Year, month, day). Title (edition) [Type of medium]. Volume(issue).

Akhter, M.P., Jiangbin, Z., Naqvi, I.R., Abdelmajeed, M., Mehmood, A., Sadiq, M.T.: Document-level text classification using single-layer multisize filters convolutional neural network. *IEEE Access* 8, 42689–42707 (2020)

12. Mrs Sujata Khedkar a, Dr. Subhash Shinde:Deep Learning and Ensemble Approach for Praise or Complaint Classification,sh Shinde, Professor, Computer Engineering Department, LTCE,Koparkhairane, Navi Mumbai, 400050, India,Dr. Subhash Shinde, Professor, Computer Engineering Department, LTCE,Koparkhairane, Navi Mumbai, 400709, India.

13. Joao Filgueiras ~*,Lu'is Barbosa*, Gil Rocha*, Henrique Lopes Cardoso*, Lu'is Paulo Reis*, Joao Pedro Machado ~ +, Ana Maria Oliveira,Complaint Analysis and Classification for Economic and Food Safety, *Laboratorio de Intelig'encia Artificial e Ci ^ enciade Computadores (LIACC) Faculdade deEngenhariadaUniversidade do Porto Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal.