Classification Of Student Query Using Machine Learning

BCSF188Z40 - PROJECT WORK PHASE-II REPORT

**Batch 2022 (Semester VIII)**

SUBMITTED BY

### Voore Saithanish 11189A264

**K. Sai Varun 11189M011**

GUIDED BY

Dr. M. Senthil Kumaran

**Associate Professor and HOD**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA**

March 2022

**Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya**

Enathur, Kanchipuram – 631 561



**BONAFIDE CERTIFICATE**

This is to certify that the PROJECT WORK PHASE-II Report entitled [**Classification of Student Query Using Machine Learning**] is the Bonafide work carried out by Mr.

/Ms. Voore Saithanish, K. SaiVarun Reg.No: 11189A264,11189M011 during the

academic year 2022.

**Dr. M. SenthilKumaran Dr. M. Senthil Kumaran**

Professor Head of the Department

Associate Department of CSE, Department of CSE,

SCSVMV SCSVMV

Submitted for the project work viva - voce examination held on 30-03-2022

Place: Kanchipuram. Date:30-03-2022

***Examiner 1 Examiner2***

**DECLARATION**

It is certified that the PROJECT WORK PHASE-II work titled **Classification of Student Query Using Machine Learning** is originally implemented by us. No ideas, processes, results or words of others have been presented as my own work. Due acknowledgement is given wherever others' work or ideas are utilized.

1. There is no fabrication of data or results which have been compiled /analyzed.
2. There is no falsification by manipulating data or processes, or changing or omitting data or results.

I understand that the project is liable to be rejected at any stage (even at a later date) if it is discovered that the project has been plagiarized, or significant code has been copied. I understand that if such malpractices are found, the project will be disqualified, and the Degree awarded itself will become invalid.

Signature of Student(s) with date: VOORE SAITHANISH [30-03-2022]

K.SAI VARUN [30-03-2022]

### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NUMBER** |
|  | Abstract  List of Figures  List of Abbreviations | I  II III |
| 1 | **Introduction**  1.1 Objectives | 1  2 |
|  | 1.2 Scope of the Project |  |
| 2 | **Literature Survey**   * 1. Literature Survey   2. Problem Statement | 3  3 |
| 3 | **Project Description**  3.1 Proposed Method | 4  6  7  9  10 |
|  | 3.2 Algoritm |
|  | 3.3 Architecture |
|  | 3.4 Methodology |
|  | 3.5 Project Description |
| 4 | **Conclusion** | 29 |
| 5 | **References** | 31 |

**Abstract**

The Educational institutions and universities were getting a bulk amount of data in the form of queries to 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 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 a 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)

### I

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGUR**  **ES NO** | **FIGURE NAME** | **PAGE NO** |
| 3.3.1 | Flow Chart | 8 |
| 3.5.1 | The Accuracy and Deviation Shown as Output | 11 |
| 3.5.2 | The Output Shows  the Sorting of Data as of Category\_id | 11 |
| 4.2.1 | The Queries Classified into Respective  Departments | 28 |
| 5.1 | Accuracy Graph | 29 |

**II**

**List Of Abbreviations:**

|  |  |
| --- | --- |
| ML | Machine Learning |
| TF-IDF | Term Frequency-Inverse Document Frequency |
| SVM | Support Vector Machine |
| CSV | Comma Separated Values |
| CAT | Category |
| DF | Data Frame |
| Multinomialnb | Multinomial Naive Bayes |
| RFC | Random Forest Classifier |

**III**

**Chapter 1 Introduction**

The data received from students to the universities in daily bias in the bulk from which makes it the universities difficult to sort out the queries according to the departments, taking a 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 more complex to find out and resolve in a period. The students facing problems as well as the period of resolving their queries is delayed too. So, this project where is designed to classify the data into the departments by giving the data keywords and making them into subgroups which the algorithm differentiates the data into types of departments that make 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 priority and the student receives the notification of his/her status of the query. The department gets informed regarding the query, time posted, a priority which makes it the department easier to resolve the query. After the query is 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 makes the task simple for the management.

# Objectives

## The main objective is to make the task easy and in a short span of time

1

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

* 1. **Scope of the Project**

The complaints received are in the form of text, to classify the complaints with the help of a classification algorithm, the text needs to be transformed into vectors so that the algorithm will be able to predict the class. To achieve it, we use the TF-IDF method to convert the text to vectors. TF- IDF means term frequency-inverse document frequency which is used to find out which terms are most relevant to a specific topic. It is a statistical metric used to evaluate how relevant a term/word is to a document in a collection of documents or a corpus TF-IDF of a word in a document is calculated with the help of two measures TF (term frequency) and IDF (inverse document frequency).

2

# Chapter 2

**Literature Survey**

* 1. **Literature Survey**

By the analysis of complaints data from the college we successfully built a model for classifying the complaints given by the students. we are using the machine learning classification algorithms like LinearSVC with the combination of TF-IDF (Term Frequency-Inverse Document Frequency) which computes the weight of each word and signifies the importance of the word in the document and corpus this method is a widely used technique in Information Retrieval and Text Mining. Our model results in an accuracy of 89% and makes every complaint system work efficiently for maintaining a huge number of complaints. More productivity of work and consuming more time will vanish and help in the growth of Institutions

# 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 quickly and easily 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 lucidly. No more confusion and complex situations such as clashing the queries and not being able to find one in a bulk file.

3

# Chapter 3

**Proposed Method / Algorithm / Architecture / Process / Methodology**

**/ Project Description**

* 1. **Proposed Method**

The complaints received are in the form of text, to classify the complaints with the help of a classification algorithm, the text needs to be transformed into vectors so that the algorithm will be able to predict the class. To achieve it, we use the TF-IDF method to convert the text to vectors. TF-IDF means term frequency-inverse document frequency which is used to find out which terms are most relevant to a specific topic. It is a statistical metric used to evaluate how relevant a term/word is to a document in a collection of documents or a corpus TF-IDF of a word in a document is calculated with the help of two measures TF (term frequency) and IDF (inverse document frequency). TF (term frequency) is calculated by finding several times a word appeared in a document and the frequency is adjusted with the length of the document or the number of words in the document. IDF (inverse document frequency) of a word or term means how rare or low the word appears in the entire corpus or collection of documents. This can be calculated by dividing the number of documents concerning the number of documents the term appeared. If the word or term appears in more no of the document and is very common, then it is scaled to ‘0’ else if it is scaled to ‘1’. Multiplying the two terms with each other we can obtain the TD-IDF score. Higher the score higher the relevancy of the word concerning the document. After converting the text, we apply the different classification algorithms like Random Forest Classifier, Linear SVC, Multinomial NB, and Logistic Regression the Dataset “complaints.csv” will be having the attributes as Token No, Date, Year, Student-ID, Email Id, Grievance Category, Counselor Name, Cat, Issue Resolver Name, Issue Given Date, No of Days to Resolve, Issue Resolved Status, Final Status. By using this “complaints.csv” dataset we will make another copy of the Data Frame consisting of

4

category in other works making a temporary dictionary for future reference. Now we can also know that which section or department is having a greater number of complaints raised by students. Now, we will be applying the TFID-Vectorizer which will transform each complaint into a vector, and we will be storing the vectors in an array and we can get the score of Unigrams and Bigrams. After we will map the vectors with the most correlated Unigrams and Bigrams for each complaint by removing the stop words. The splitting of data for Training and Testing will take place like ‘X’ which is having all the Grievance Category and ‘y’ which is consisting of the target labels we need to predict By this step, everything will be sorted out with training and testing the data. Now we apply different machine learning classification algorithms and predict the output for the given complaints. Now the other part of the project is maintaining the database for sending the notification bidirectional regarding the complaints. For that when the classification process is completed the predicted output will be taken and based on that prediction, we will trigger the notification for that department employee who will be resolving the complaint. Finally, when the

complaint is resolved and once updated on the website the resolved notification will be triggered back to the issue raiser and work will be completed easily without any wasting of time it will be best when compared to all complaint classifiers as it is a one-to-one interaction.

5

# Algorithm

### Input:

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

Weight Matrix (which comprises of the multitude of loads of terms that are called vectors)

### Method:

1-for every grumbling archive (ci) do 2-for each term (tj) in ci do

1. 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) = log2 ((Total Documents)/ (records With term tj))

1. End for of term
2. End for of objection record
3. The vectors are put away in an exhibit for preparing and testing purposes, during arrangement.

6

# Architecture

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.

7

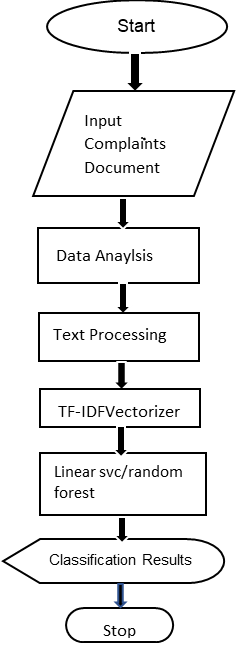


Fig: 3.3.1

Flow Chart

8

# Methodology

The issues which were stored in the database within one or two days the exact solution will be given to person whether his/her issue was resolved successfully or not. The classification algorithms combining with TF-IDF(term frequency-inverse document frequency) based on Text Classification which we were using classifies the issue based on the frequency of the works in the given complaint and making vectors by converting the text and divide it into a particular domain and maps the complaint to respective department and role of classification algorithms takes place where we can categorize the data to a particular department based on the labels given pre-definitely for each respective departments and sends a notification to that department admin. The main of this project say that the person should not lose his valuable time and make the work easier by using different machine learning algorithms. So, in this way, we can design, that the person can raise a complaint and get the solution to his/her complaint easier. The web system using technologies is the easiest way to solve the complaints raised by the students in bulk amount. Hence using this developed model, the complaints can be solved easier and faster by classifying to required departments of the Organization.

9

# 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

10

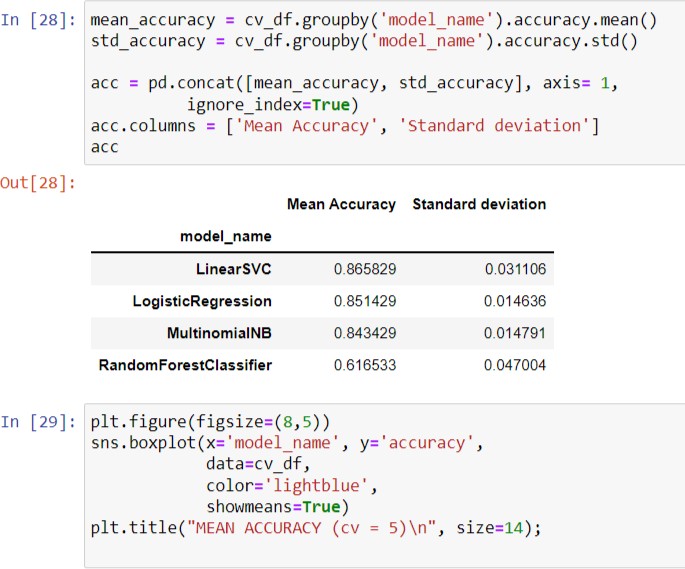


Fig: 3.5.1

The Accuracy and Deviation Shown as Output

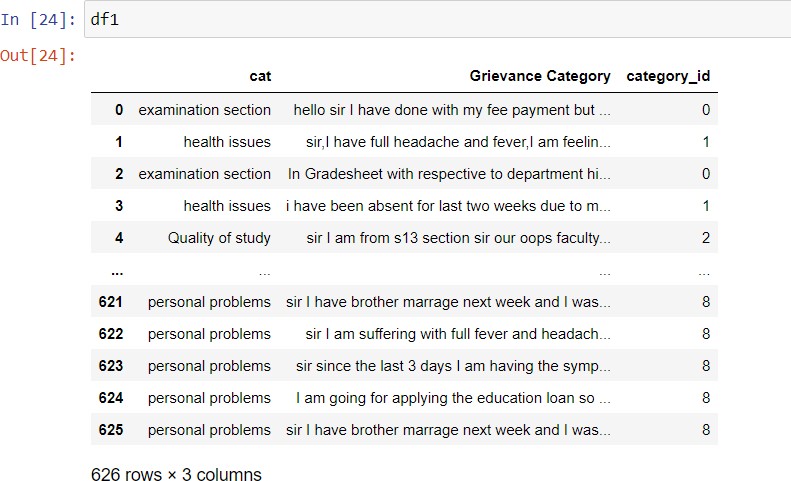


Fig: 3.5.2

The output shows the sorting of data as of category\_id

11

# Chapter 4

**Implementation / Code / Results and Description of Results**

* 1. **Code**

4/2/2021

projectreview - Jupyter Notebook

In [2]: import numpy as np In [3]: import pandas as pd

In [4]: df=pd.read\_csv(r"C:\Users\Sheri Sumanth Reddy\Desktop\complaintsv3.csv",encoding= 'unicode\_escape')

localhost:8888/notebooks/projectreview.ipynb 1/12

4/2/2021

projectreview - Jupyter Notebook In [5]: df

Out[ 5 ] :

Token No

12

Date Year Student ID Email

Grievance Category

Counsell‹

0 51003 10/5/2020 3 180030337 [11179a2002@kanchiuniv.ac.in](mailto:11179a2002@kanchiuniv.ac.in) I have search the

book In the Dr.A.Praveen Iibrary,but it localhost:8888/notebooks/projectreview.ipynb sir,at the time of registration I chosen wrong...

I have taken a book,but the book was replaced ...

In Gradesheet with respective to department hi...

I am unable to publish the paper related to my...

I have taken a book,and I have not returned th...

13

sir,I am veeresh bearing id no 170030691.

I am...

sir,I am from s1 section even after paying sem...

Token No

projectreview - Jupyter Notebook

Date Year Student ID Email Grievance Counsell‹

I have more than 90

3127 241778 8/25/2020 4 170030411 [11179a266@kanchiuniv.ac.in](mailto:11179a266@kanchiuniv.ac.in) percentage

attendence in D...

3128 rows x 15 columns

In [6]: df1 = df[['cat', 'Grievance Category']].copy()\

In [7]:

Out [7 ] :

df1 cat

14

Grievance Category

1. library issues I have search the book In the Iibrary,but it i...
2. registrations sir,at the time of registration I chosen wrong...
3. library issues I have taken a book,but the book was replaced ...
4. examination section In Gradesheet with respective to department hi...
5. project I am unable to publish the paper related to my...

3123 library issues I have taken a book,and I have not returned th... 3124 examination section sir I am from cse department and I have lost m... 3125

health issues sir,I am veeresh bearing id no 170030691. I am... 3126 examination sir,I am from s1 section even after paying sem... 3127 detention I have

more

than 90 percentage attendence in D... 3128 rows • 2 columns

In [8]:

Out [8 ] :

df1 = df1[pd.notnull(df1['Grievance Category'])]

# Renaming second column for a simpler name df1.columns = ['cat', 'Grievance Category']

df1.shape ( 3128, 2)

localhost:8888/notebooks/projectreview.ipynb

3/12

15

projectreview - Jupyter Notebook

In [9]: df1 = df1[pd.notnull(df1['Grievance Category'])]

# Renaming second column for a simpler name df1.columns = ['cat', 'Grievance Category'] df1.shape

Out[9]:

In [10]:

['category\_id'] = df1['cat'].factorize()[0]

category\_id\_df = df1[['cat', 'category\_id']].drop\_duplicates() df1

# Dictionaries for future use

category\_to\_id = dict(category\_id\_df.values)

id\_to\_category = dict(category\_id\_df[['category\_id', 'cat']].values) # New dataframe

df1.head() Out[10]:

df1.head()

cat Grievance Category category\_id

1. library issues I have search the book In the Iibrary,but it i... 0
2. registrations sir,at the time of registration I chosen wrong... 1
3. library issues I have taken a book,but the book was replaced ... 0
4. examination section In Gradesheet with respective to department hi...
5. project I am unable to publish the paper related to my...

3

16

In [11]:

Out[11]:

In [12]:

df1.category\_id.unique()

array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], dtype=int64)

import numpy as np

from scipy.stats import randint

import seaborn as sns # used for plot interactive graph. import matplotlib.pyplot as plt

import seaborn as sns from io import StringIO

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.feature\_selection import chi2 from IPython.display import display

from sklearn.model\_selection import train\_test\_split

from sklearn.feature\_extraction.text import TfidfTransformer from sklearn.naive\_bayes import MultinomialNB

from sklearn.linear\_model import LogisticRegression from sklearn.ensemble import RandomForestClassifier

from sklearn.svm import LinearSVCfrom sklearn.metrics import confusion\_matrix from sklearn import metrics

localhost:8888/notebooks/projectreview.ipynb In [13]:

fig = plt.figure(figsize=(8,6))

colors = ['grey','grey','grey','grey','grey','grey','grey','grey','grey',

17

'grey','darkblue','darkblue','darkblue'] df1.groupby('cat').category\_id.count().sort\_values().plot.barh(

ylim=0, color=colors, title= 'NUMBER OF COMPLAINTS IN EACH PRODUCT CATEGORY\n')

plt.xlabel('Number of ocurrences', fontsize = 10);

localhost:8888/notebooks/projectreview.ipynb In [ 14] : df-1

cat

0 library issues Grievance Category

I have search the book In the Iibrary,but it i... category\_id

0

1. registrations sir,at the time of registration I chosen wrong... 1
2. library issues I have taken a book,but the book was replaced ... 0
3. examination section In Gradesheet with respective to department hi... 2
4. project I am unable to publish the paper related to my... 3

3123 library issues I have taken a book,and I have not returned th...

18

|  |  |  |
| --- | --- | --- |
| 3124 | examination section | sir I am from cse department and I have lost m... |
| 2 |  |  |
| 3125 | health issues | sir,I am veeresh bearing id no 170030691. I am... |
| 10 |  |  |
| 3126 | examination section | sir,I am from s1 section even after paying sem... |
| 2 |  |  |
| 3127 | detention | I have more than 90 percentage attendence in D... |

3128 rows • 3 columns

In [15]:

tfidf = TfidfVectorizer(sublinear\_tf=True, min\_df=5, ngram\_range=(1, 2), stop\_words='english')

# We transform each complaint into a vector

features = tfidf.fit\_transform(df1['Grievance Category']).toarray() labels = df1.category\_id

print("Each of the %d complaints is represented by %d features (TF-IDF score of unigrams and bigrams)" %(features.shape))

In [ 16] :

N = 4

for cat, category\_id in sorted(category\_to\_id.items()):

features\_chi2 = chi2(features, labels == category\_id) indices = np.argsort(features\_chi2[0])

feature\_names = np.array(tfidf.get\_feature\_names())[indices] unigrams = [v for v in feature\_names if len(v.split(' ')) == 1]

19

bigrams = [v for v in feature\_names if len(v.split(' ')) == 2] print("\n==> %s:" %(cat))

print(" \* Most Correlated Unigrams are: %s" %(', '.join(unigrams[-N:]))) print(" \* Most Correlated Bigrams are: %s" %(', '.join(bigrams[-N:])))

==> Quality of study:

* Most Correlated Unigrams are: topics, ppts, faculty, explaining
* Most Correlated Bigrams are: technologies text, faculty members, faculty ppts

==> certifications:

* Most Correlated Unigrams are: certification, rpa, completed, course
* Most Correlated Bigrams are: rpa course, course certificate, completed rp a, certificate

generated

==> detention:

* Most Correlated Unigrams are: detention, marks, person, detained
* Most Correlated Bigrams are: detention list, detained subjects, sir detai ned, detained subject

==> examination section:

* Most Correlated Unigrams are: hallticket, grade, memo, gradesheet
* Most Correlated Bigrams are: examination fee, hallticket generated, fee h allticket, gradesheet

respective

20

==> facilities issues:

* Most Correlated Unigrams are: allocated, food, students, chairs
* Most Correlated Bigrams are: waiting long, food schedule, provided studen ts, classroom provided

==> health issues:

* Most Correlated Unigrams are: absent, week, weeks, rest
* Most Correlated Bigrams are: low resolve, issues attendence, absent week s, attendence low

==> job/placements issues:

* Most Correlated Bigrams are: companies received, letter till, company did n, mnc company

==> library issues:

* Most Correlated Unigrams are: taken, books, book, library
* Most Correlated Bigrams are: library online, took books, book library, ta ken book

==> non-technical:

* Most Correlated Unigrams are: dance, kits, nss, member
* Most Correlated Bigrams are: making big, need work, project need, member nss

==> personal problems:

* Most Correlated Unigrams are: area, marriage, brother, meeting
* Most Correlated Bigrams are: days climate, area permission, attend brothe r, brother marriage

21

==> project:

* Most Correlated Unigrams are: paper, saying, guide, project
* Most Correlated Bigrams are: change date, unable complete, project given, deadline change

==> registrations:

* Most Correlated Unigrams are: updated, sem, erp, registration
* Most Correlated Bigrams are: erp resolve, send updated, sem fee, time reg istration

==> schedules:\* Most Correlated Bigrams are: gre coaching, classes gre, class time, chang e schedule

In [17]: df1 Out[17]:

cat Grievance Category

1. library issues I have search the book In the Iibrary,but it i...
2. registrations sir,at the time of registration I chosen wrong...
3. library issues I have taken a book,but the book was replaced ...
4. examination section In Gradesheet with respective to department hi... category\_id

0

1

0

2

1. project I am unable to publish the paper related to my... 3

22

|  |  |  |
| --- | --- | --- |
| 3123 | library issues | I have taken a book,and I have not returned th... |
|  | 0 |  |
| 3124 | examination section | sir I am from cse department and I have lost m... |
| 2 |  |  |
| 3125 | health issues | sir,I am veeresh bearing id no 170030691. I am... |

3126 examination section sir,I am from s1 section even after paying sem...

2

3127 detention I have more than 90 percentage attendence in D...

5

3128 rows • 3 columns

In [ 18] :

X = df1['Grievance Category'] # Collection of documents

y = df1['cat'] # Target or the labels we want to predict (i.e., the 13 different complaints of products)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size=0.25, random\_state = 0)

In [ 19] :y\_train Out [ 19] :

In [20]:models = [

RandomForestClassifier(n\_estimators=100, max\_depth=5, random\_state=0), LinearSVC(),

MultinomialNB(), LogisticRegression(random\_state=0),

]

23

# 5 Cross-validation

CV = 5

cv\_df = pd.DataFrame(index=range(CV \* len(models))) model\_name = model. class . name

accuracies = cross\_val\_score(model, features, labels, scoring='accuracy', cv=CV) for fold\_idx, accuracy in enumerate(accuracies):

entries.append((model\_name, fold\_idx, accuracy))

cv\_df = pd.DataFrame(entries, columns=['model\_name', 'fold\_idx', 'accuracy']) In [21] :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[21]:

|  |  |
| --- | --- |
| 2741 | health issues |
| 2649 | project |
| 1236 | registrations |
| 3111 | examination section |
| 2485 | health issues |

763 detent1on

835 health issues

|  |  |
| --- | --- |
| 1653 | non-technical |
| 2607 | Quality of study |
| 2732 | health issues |

24

LinearSVC LogisticRegression MultinomialNB RandomForestClassifier localhost:8888/notebooks/projectreview.ipynb

0.983058

0.983058

0.984977

0.696927

0.005481

0.005481

0.006735

0.005843

9/12

4/2/2021

projectreview - Jupyter Notebook In [22]:

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) In [23]:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y,

test\_size=0.25,

tfidf = TfidfVectorizer(sublinear\_tf=True, min\_df=5, ngram\_range=(1, 2), stop\_words='english')

fitted\_vectorizer = tfidf.fit(X\_train)

25

tfidf\_vectorizer\_vectors = fitted\_vectorizer.transform(X\_train) model = LinearSVC().fit(tfidf\_vectorizer\_vectors, y\_train) localhost:8888/notebooks/projectreview.ipynb

10/12

4/2/2021

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 timeable 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 coursera certification in DSLR 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 a issue with my hostel food from the last week"""

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 Samyak so I need permission for making announcements in classes"""

26

new\_complaint12 = """Mam, 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 changing the details"""

new\_complaint14 = """I have search the book In the library,but it is out of stock in the library"""

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])))

In [27]:import joblib In [36]:import joblib

27

# Result

By applying the different classification algorithms like Random Forest Classifier, Linear SVC, Multinomial NB and Logistic Regression for getting different predictions for each classification algorithm and produced the mean accuracy of 0.61, 0.86, 0.84,

0.84 and standard deviation of 0.052, 0.031, 0.019, 0.016 respectively for the classification algorithms. After cross-validation is processed for evaluating the accuracies and storing them in a separate Data Frame for further references. By our work, we have chosen Linear SVC as the classification algorithm which is producing more accurate results for the classification process and we are getting an accuracy of 89% for our Dataset.

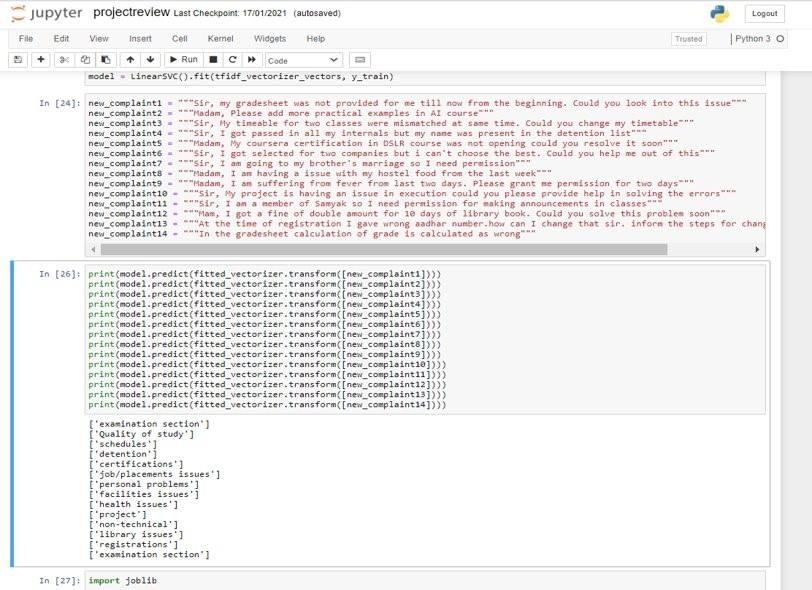


Fig: 4.2.1

The Queries Classified into Respective Departments

28

# Chapter 5 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.

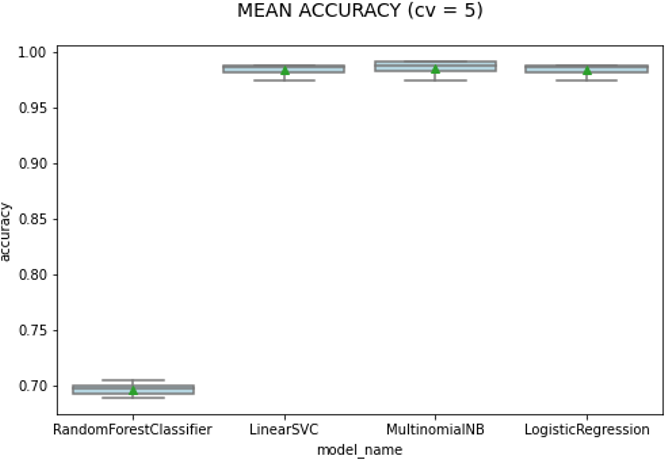


Fig: 5.1

Accuracy Graph

29

**References**

1. N. S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3):175–185.
2. Koray Balcı -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,
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, Brawijiaya 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.

30

1. 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.
2. 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)

1. 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.
2. Joao Filgueiras ˜ \*,Lu´ıs Barbosa\*, Gil Rocha\*, Henrique Lopes Cardoso\*, Lu´ıs Paulo Reis\*, Joao Pedro Machado ˜ +, Ana Maria Oliveira,Complaint Analysis and Classification for Economic and Food Safety, \*Laboratorio de Intelig ´ encia Artificial e Ci ˆ enciade Computadores (LIACC) Faculdade deEngenhariadaUniversidade do Porto Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal.

31

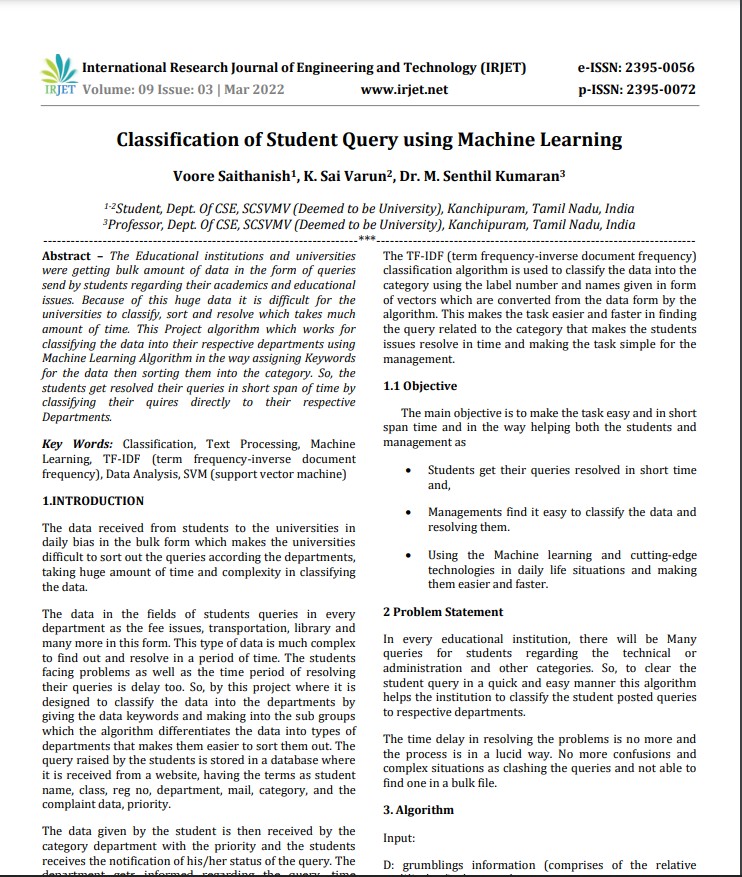
# Publication Details and Proofs

**1. Journal name:** International Research Journal of engineering and technology

**2. DOI: e-ISSN: 2395-0056 p-ISSN: 2395-0072**

**3**. **Publication Link:** <https://www.irjet.net/volume9-issue3>

## International Research Journal of Engineering and Technology (IRJET) Volume 9, Issue 3, March 2022



32