

SERVICE REQUEST CLASSIFICATION BASED ON PRIORITY

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

BALAM KARTHIK YADAV
(Reg. No.37110072)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC JEPPIAAR NAGAR, RAJIV
GANDHI SALAI, CHENNAI – 600 119

MARCH – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **BALAM KARTHIK YADAV (Reg. No.37110072)** who carried out the project entitled "**SERVICE REQUEST CLASSIFICATION BASED ON PRIORITY**" under my supervision from November 2020 to March 2021.

Internal Guide

Dr. N. Srinivasan, M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **AYITHA VENKATA SUBBA MANIKANTA** hereby declare that the Project Report entitled “**SERVICE REQUEST CLASSIFICATION BASED ON PRIORITY**” is done by me under the guidance of **Dr. N. Srinivasan, M.E.,Ph.D.**,Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. N. Srinivasan, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Real-time service request management isn't required just for emergency responses. This is the next evolution of service request management and it will become the norm for service request management expectations, service levels and experience levels. Organizations that invest in these capabilities early could gain an advantage over their competitors that don't. After all, the customer is king because he or she expects excellent service support and the best product isn't always the winner if the customer is frustrated. Business disruptions are costly to both the supplier and the customer of the service. Business as usual must resume as soon as possible to minimize disruption and maybe even to save lives. Disruption is not always defined as downtime of a system, but may also be a service incapable of functioning normally. This could indicate performance issues related to accessing and utilizing the service.

TABLE OF CONTENTS

	ABSTRACT	v
	LIST OF FIGURES	viii
CHAPTER NO	TITLE	PAGE No
1	INTRODUCTION.	1
2	LITERATURE SURVEY.	2
3	AIM AND SCOPE.	13
	3.1 AIM OF PROJECT.	13
	3.1.1 OBJECTIVES.	13
	3.2 SCOPE.	13
	3.2.1 ADVANTAGES.	13
4	SYSTEM DESIGN & METHODOLOGY.	14
	4.1 EXISTING SYSTEM.	14
	4.2 PROPOSED SYSTEM.	14
	4.2.2 ADVANTAGES OF PROPOSED SYSTEM.	15
	4.3 THE STAGES IN THE INCIDENT MANAGEMENT PROCESS.	15
	4.4 ARCHITECTURE DIAGRAM.	16
	4.5 SYSTEM REQUIREMENTS.	17
	4.5.1 HARDWARE REQUIREMENTS.	17
	4.5.2 SOFTWARE REQUIREMENTS.	17
	4.6 SOFTWARE ENVIRONMENT.	17
	4.6.1 PYTHON.	17
	4.6.2 HISTORY OF PYTHON.	18
	4.6.3 PYTHON FEATURES.	18
	4.6.4 GETTING PYTHON.	19
	4.7 ALGORITHM.	21
	4.7.1 RANDOM FOREST ALGORITHM.	21
	4.7.2 WORKING OF RANDOM FOREST.	21

	4.8 MODULES.	22
	4.8.1 SERVICE REQUEST IDENTIFICATION.	22
	4.8.2 SERVICE REQUEST LOGGING.	22
	4.8.3 SERVICE REQUEST CATEGORIZATION.	23
	4.8.4 SERVICE REQUEST PRIORITIZATION.	23
	4.9 DATA FLOW DIAGRAM.	24
5	RESULT AND DISCUSSION.	26
6	CONCLUSION.	29
	REFERENCES	30
	APPENDIX	
	A. PLAGIARISM REPORT	31
	B. SOURCE CODE	32

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.4.1	ARCHITECTURE DIAGRAM.	16
4.9.1	DATA FLOW DIAGRAM.	25
5.1	SERVICE REQUEST LOGGING.	26
5.2	SERVICE REQUEST CATEGORIZATION.	27
5.3	SERVICE REQUEST PRIORITIZATION.	28
5.4	ACCURACY SCORE.	28

INTRODUCTION

INTRODUCTION

Service composition encompasses all those processes that create added-value services, called composite or aggregated services, from existing services. The need to integrate services is recognized in both the enterprise and the consumer arenas. A Gartner report stated that “through 2014, the act of composition will be a stronger opportunity to deliver value from software than the act of development”. More recently the focus has shifted to more specific types of compositions: the “Top 10 Strategic Technology Trends for 2014” Gartner report confirms the significance of composition of cloud services, and Forrester in its “2015 Predictions for Application Development and Delivery Professionals” highlights composition for mobile apps (i.e., assembling front-end components) as a cornerstone to enhance development productivity, leveraging Web components and platforms such as HTML5 and Google Polymer, a library for creating these components.

For a company that offers services and solution for customers, a reliable Service request classification based on priority is essential. Some customers even demand it. Customers need to easily be able to contact the service provider if any problems occur. On the company side they could also provide valuable information. For example statistics on how many incidents customers report regarding specific services or subjects. A big issue when handling incidents from different companies is security. This report will describe one way, the best way within the provided timeframe, of how to implement a complete service request classification based on priority in SharePoint. Some challenges we will face include: how to develop web parts for SharePoint since that is a completely new development environment for us, how to secure information without severe performance hits, and how to gather and present statistics about incident reports. The project touches several subjects in computer science: CSV files as data sets, Python, Machine learning algorithms.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
1	Web service composition: A survey of techniques and tools.	A.L. Lemos, F. Daniel, and B. Benatallah	Service composition is a complex task that has inspired a variety of automation techniques trying to overcome someof the complexity. We identified three automation techniquesof major importance synthesis, planning, and model-driven development.	Clearly, previous efforts by research and practitioner Communities have produced promising results that are certainly useful.	A commonly overlooked limitation of current systems is that they do not make composition languages accessible to end users.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
2	A Survey on issues and Enabling Technologies	Anne H.H.Ngu, Mario Gutierrez, Vangelis Metsis, Surya Nepal, and Quan Z.Sheng, Member, IEEE	We believe emerging techniques such as privacy by design, differential privacy, and light weight public key cryptography will form the building blocks for security in IOT middle ware.	The results from these systems are still experimental and preliminary. in[46], Ostermaier al. Reported a system, named Dyser, for real-time discovery of things on the Web	Limitations of CPU and memory put sever restrictions on the utility of the resource-demanding public-key cryptography as used in the internet security.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
3	A survey on enabling technologies, protocols and applications.	A. Al-Fuqaha, M.Guizani, M.MohammadiM.Aledhari, and M.Ayyash.	We also reviewed the role of cloud and fog computing in the IoT ecosystem. At the cloud level, we need platforms to support lot big data, IoT analytics and availability.	Their results show that the expected number of queries for tag identification using the : CDMA technique is lower than the EPC Gen-2 protocol.	The limitation of the cloud which offers just a specific application protocol is a barrier to its expansion.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
4	TITLEA survey on datacentric Internet of Things.	J.Y. Qin, Q. ZSheng, N. J.G. Falkner, S.Dustdar, H.Wang, and A. V.Vasilakos.	Traffic engineering techniques for managing VNFs in a SDN-based network Include for example steering network traffic through an existing sequence of VNFs by using middleboxes that modify packet headers and alter traffic signatures.	In conclusion, the lack of a standard definition of a smart city eventually results in various shortcomings in many facets of a smart city.	We highlight the achievements in realizing various aspects of smart cities, present the lessons learnt and identify limitations and research challenges.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
5	Automatic Target driven Ontology Merging.	Salvatore Raunich, Erhard Rahm	The idea is that these relationships produce different edges in the integrated concept graph and consequently different concepts and relationships in the merged taxonomy.	More recent work on schema integration builds on the research results on semiautomatic schema matching and separate matching from merging. Hence, several algorithms have been proposed to merge schemas based on a pre-determined match mapping.	The fact that different users will observe quite different QoS of the same Web service limits the applicability of this dataset.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
6	Distributed QoS valuation for RealWorld WebServices.	Zibin Zheng, Yilei Zhang, and Michael R. Lyu.	Web services have been emerging in recent years and are by now one of the most popular techniques for building versatile distributed systems.	Comprehensive experimental results are presented and reusable datasets are released.	The fact that different users will observe quite different QoS of the same Web service limits the applicability of this dataset.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
7	Bootstrapping Ontologies for Web Services	Aviv Segev, and Quan ZHI. Sheng	Our model analyzes the concept results and concept relations and performs stemming on the results. It should be noted that other techniques of clustering could be used to limit the Ontology expansion	clusters the results according to textual descriptors, and classifies which set of descriptors identifies the context of the web service.	This limitation could be overcome by either recalculating the TF and IDF after each new web service or alternatively collecting an additional set of services and reevaluating the IDF values

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
8	'Leveraging Formal concept analysis with topic correlation for service clustering and disvovery	Mustapha AZNAG, Mohamed QUAFAROU and Zahi JARIR	topic proportions exhibit correlation via the logistic normal distribution. In this paper, we utilized CTM to extract latent factors and the correlation between these topics to propose an efficient Web Service	The results show that the hierarchical clustering method based on CTM performs better than HAC. We also evaluated our service discovery approach by calculating the P recision@n and NDCGn.	In our approach, we use the Formal Concept Analysis (FCA) [3] formalism to organize the constructed hierarchical clusters into concept lattices according to their topics. Thus, service discovery may be achieved more easily using the concept lattice

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
9	Web service interface decomposition using formal concept analysis	M. Daagi, A. Ouniy, M. Kessentini, M. M. Gammoudi, and S. Bouktif	conceptual clustering technique as it also provides intensional descriptions for the abstract concepts or data units it produces. Central to FCA is the notion of a formal context:	This practice results in a "hodge-podge" of unrelated operations in a single Web Service interface. Indeed, recent studies found that developers seem to take little care of the structure of their WSDL documents.	Limitations of both approaches could be related to the coupling between interfaces which is not considered. Our approach attempts to explicitly address these two drawbacks to improve the interface design quality, taking the advantages of FCA.

S.NO	TITLE	AUTHOR	TECHNIQUES	RESULTS	LIMITATION
10	Structural and Role Oriented Web Service Discovery with Taxonomies in OWL-S	Georgios Meditskos and Nick Bassiliades	We combine object based structural matching techniques that are used in the domain of Structural Case-based Reasoning (SCBR) [6], with Description Logic (DL) reasoning [7] over Profile instances, enhancing the discovery with services that cannot be retrieved using only logic-based reasoning	he total ordering of the matches is defined based on the relaxation of the 16 grouping filters. We argue that such an approach results in a better Top-k precision	In order to overcome this limitation of the SP paradigm, we enhance our framework with the ability of a role-based Web service functional annotation based on the open world assumption and the classification capabilities of the DL reasoning paradigm

CHAPTER 3

AIM AND SCOPE

3.1 AIM OF PROJECT

The primary aim of this proposed system i.e “Service Request Classification Based on Priority” is to classify the service requests by the customers and prioritize them according to the intensity of the service request. This helps to solve the emergency requests fastly.

3.1.1 OBJECTIVES

The following are the project objectives: To develop a service solution planning considering priori knowledge that allows the user to view customers’ data as well as houses record:

- To develop a system that allows the users to add, edit, search and delete data from the database
- To study and analyse the requirement specifications of the housing society management system
- To produce the Software Requirement Specification of the system
- To produce the Software Design Document of the system

3.2 SCOPE

The project scope defines the description of the work that is required in delivering the service solution planning considering priori knowledge. The following are the scopes of work during the course of the project:

- Study and understand the requirement of this project
- Construct Software Requirement Specification document of the system
- Construct Software Design Document of the system

3.2.1 ADVANTAGES

- Moving resolution capabilities closer to the customer.
- Reduce business impact with agile, quick responses.

CHAPTER 4

SYSTEM DESIGN & METHODOLOGY

4.1 EXISTING SYSTEM

Real-time incident management helps reduce the lag when responding to a performance issue and improves the resolution time of a service not working. The customer and the supplier of the service benefit significantly from the overall experience when the distance from a performance perspective to the affected service and the capabilities used to restore the service to normal is shorter and leaner. Real-time incident management can be defined as an immediate, actionable response to an incident without the delay in complicated backend processing, which may be mostly enabled with manual resources and activities. Real-time incident management should also be more intuitive and customer-friendly to enable an exceptional experience for the user.

4.2 PROPOSED SYSTEM

- Pay attention to and analyze feedback from process users and from the customer perspective to enable agility.
- Practice and visualize processes with your staff, using technology-use cases. This helps to improve performance when real-time incident management is needed.
- Practice using the technology in a self-help manner from the customer perspective to understand the customer experience.
- Formally define process, procedures and work instructions and add a knowledge base for easy access. Many organizations do not do this, which results in questions that may require just-in-time decision-making. Failing to do this also contributes to inconsistent actions related to real-time incident management.
- Invest in emerging technology to support the goals of incident management, which will lead to real-time incident management capabilities.

Speed and performance of incident resolution have always been a priority for many organizations, so they can serve their customers better. To be more effective at this task,

incident management capabilities must be moved from the service desk analyst and closer to supporting the customers of the service.

4.2.2 ADVANTAGES OF PROPOSED SYSTEM

- Moving resolution capabilities closer to the customer.
- Reduce business impact with agile, quick responses.

4.3 THE STAGES IN THE INCIDENT MANAGEMENT PROCESS

Incident Management process encompasses the following sequence of actions:

- Incident logging
- Incident Classification
- Incident Prioritization
- Investigation and Diagnosis
- Incident resolution & closure

4.4 ARCHITECTURE DIAGRAM

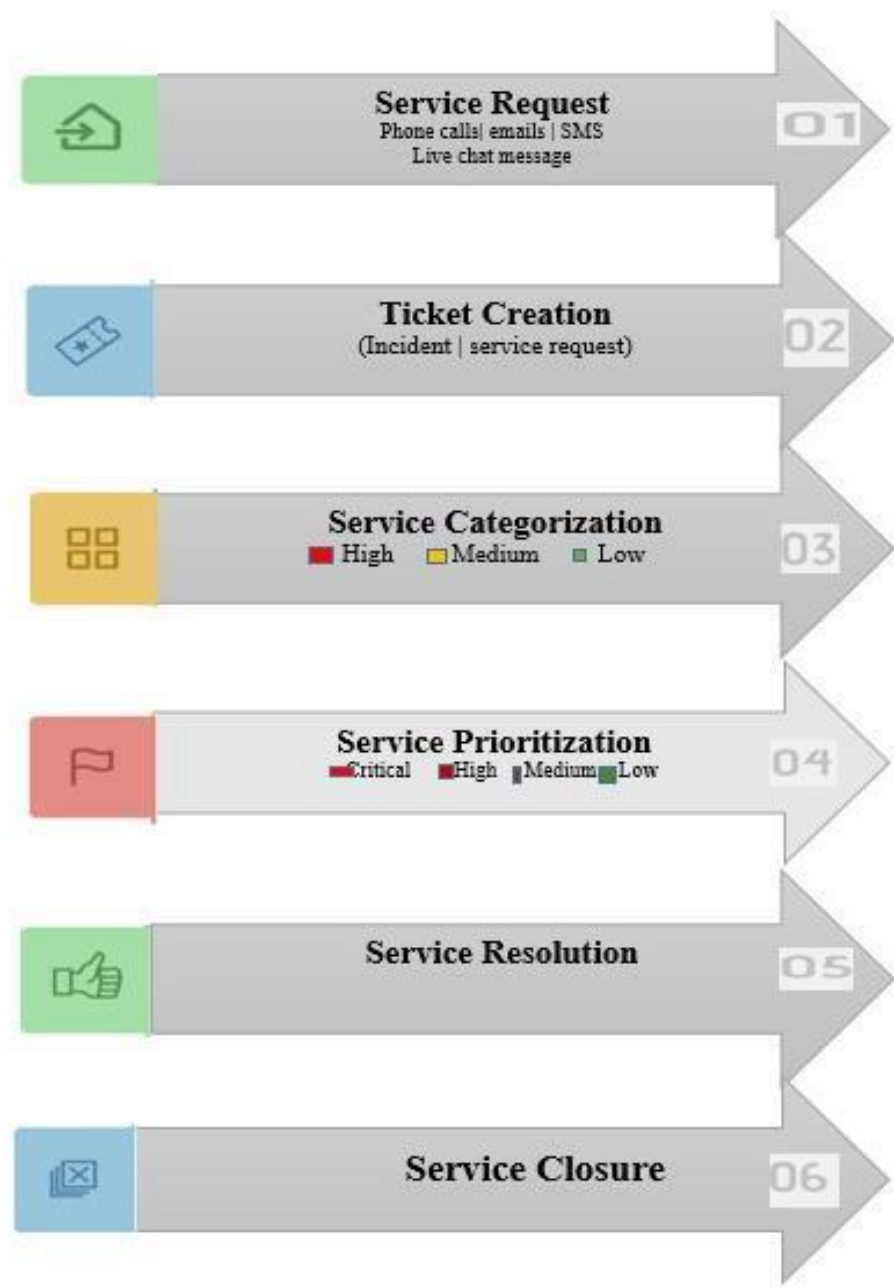


FIG 4.4.1 ARCHITECTURE DIAGRAM

4.5 SYSTEM REQUIREMENTS

4.5.1 HARDWARE REQUIREMENTS

- System : Pentium Dual Core.
- Hard Disk : 120 GB.
- Monitor : 15” LED
- Input Devices : Keyboard, Mouse
- Ram : 4 GB.

4.5.2 SOFTWARE REQUIREMENTS

- Operating system: Windows 7/10.
- Coding Language :Python, Machine Learning algorithms

4.6 SOFTWARE ENVIRONMENT.

4.6.1 PYTHON:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.6.2 HISTORY OF PYTHON:

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

4.6.3 PYTHON FEATURES:

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.

- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

4.6.4 GETTING PYTHON:

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>. Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.
- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help","copyright","credits"or"license"for more information.

>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!");**. However in Python version 2.4.3, this produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

4.7 ALGORITHM

4.7.1 RANDOM FOREST ALGORITHM:

- Random forest is a supervised classification machine learning algorithm which uses ensemble method. Simply put, a random forest is made up of numerous decision trees and helps to tackle the problem of overfitting in decision trees. These decision trees are randomly constructed by selecting random features from the given dataset. Random forest arrives at a decision or prediction based on the maximum number of votes received from the decision trees. The outcome which is arrived at, for a maximum number of times through the numerous decision trees is considered as the final outcome by the random forest.

4.7.2 WORKING OF RANDOM FOREST:

- Random forests are based on ensemble learning techniques. Ensemble, simply means a group or a collection, which in this case, is a collection of decision trees, together called as random forest. The accuracy of ensemble models is better than the accuracy of individual models due to the fact that it compiles the results from the individual models and provides a final outcome.
- How to select features from the dataset to construct decision trees for the Random Forest?
- Features are selected randomly using a method known as bootstrap aggregating or bagging. From the set of features available in the dataset, a number of training

subsets are created by choosing random features with replacement. What this means is that one feature may be repeated in different training subsets at the same time.

4.8 MODULES.

- Service request identification.
- Service request logging.
- Service request categorization.
- Service request prioritization.
- Service request response.

4.8.1 SERVICE REQUEST IDENTIFICATION.

- The first step in the life of an Service request is Service request identification. Service requests come from users in whatever forms the organization allows.
- Sources of Service request reporting include walk-ups, self-service, phone calls, emails, support chats, and automated notices, such as network monitoring software or system scanning utilities.
- The service desk then decides if the issue is truly an Service request or if it's a request. Requests are categorized and handled differently than Service requests, and they fall under request fulfillment.

4.8.2 SERVICE REQUEST LOGGING

- Once identified as an Service request, the service desk logs the Service request as a ticket.
- The ticket should include information, such as the user's name and contact information, the Service request description, and the date and time of the Service request report (for SLA adherence).

- The logging process can also include categorization, prioritization, and the steps the service desk completes.

4.8.3 SERVICE REQUEST CATEGORIZATION.

- Categorization involves assigning a category and at least one subcategory to the Service request. This action serves several purposes.
- First, it allows the service desk to sort and model Service requests based on their categories and subcategories.
- Second, it allows some issues to be automatically prioritized. For example, an Service request might be categorized as “network” with a sub-category of “network outage”.
- The third purpose is to provide accurate Service request tracking. When Service requests are categorized, patterns emerge. It’s easy to quantify how often certain Service requests come up and point to trends that require training or problem management. For example, it’s much easier to sell the CFO on new hardware when the data supports the decision.

4.8.4 SERVICE REQUEST PRIORITIZATION

- Service request prioritization is important for SLA response adherence. An Service request’s priority is determined by its impact on users and on the business and its urgency. Urgency is how quickly a resolution is required; impact is the measure of the extent of potential damage the Service request may cause.
- **Low-priority Service requests** are those that do not interrupt users or the business and can be worked around. Services to users and customers can be maintained.
- **Medium-priority Service requests** affect a few staff and interrupt work to some degree. Customers may be slightly affected or inconvenienced.

- **High-priority Service requests** affect a large number of users or customers, interrupt business, and affect service delivery. These Service requests almost always have a financial impact.
- **Service request response**
 - Once identified, categorized, prioritized, and logged, the service desk can handle and resolve the Service request. Service request resolution involves five steps:
 - **Initial diagnosis:** This occurs when the user describes his or her problem and answers troubleshooting questions.
 - **Service request escalation:** This happens when an Service request requires advanced support, such as sending an on-site technician or assistance from certified support staff. As mentioned previously, most Service requests should be resolved by the first tier support staff and should not make it to the escalation step.
 - **Investigation and diagnosis:** These processes take place during troubleshooting when the initial Service request hypothesis is confirmed as being correct. Once the Service request is diagnosed, staff can apply a solution, such as changing software settings, applying a software patch, or ordering new hardware.
 - **Resolution and recovery:** This is when the service desk confirms that the user's service has been restored to the required SLA level.
 - **Service request closure:** At this point, the Service request is considered closed and the Service request process ends.

4.9 DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail

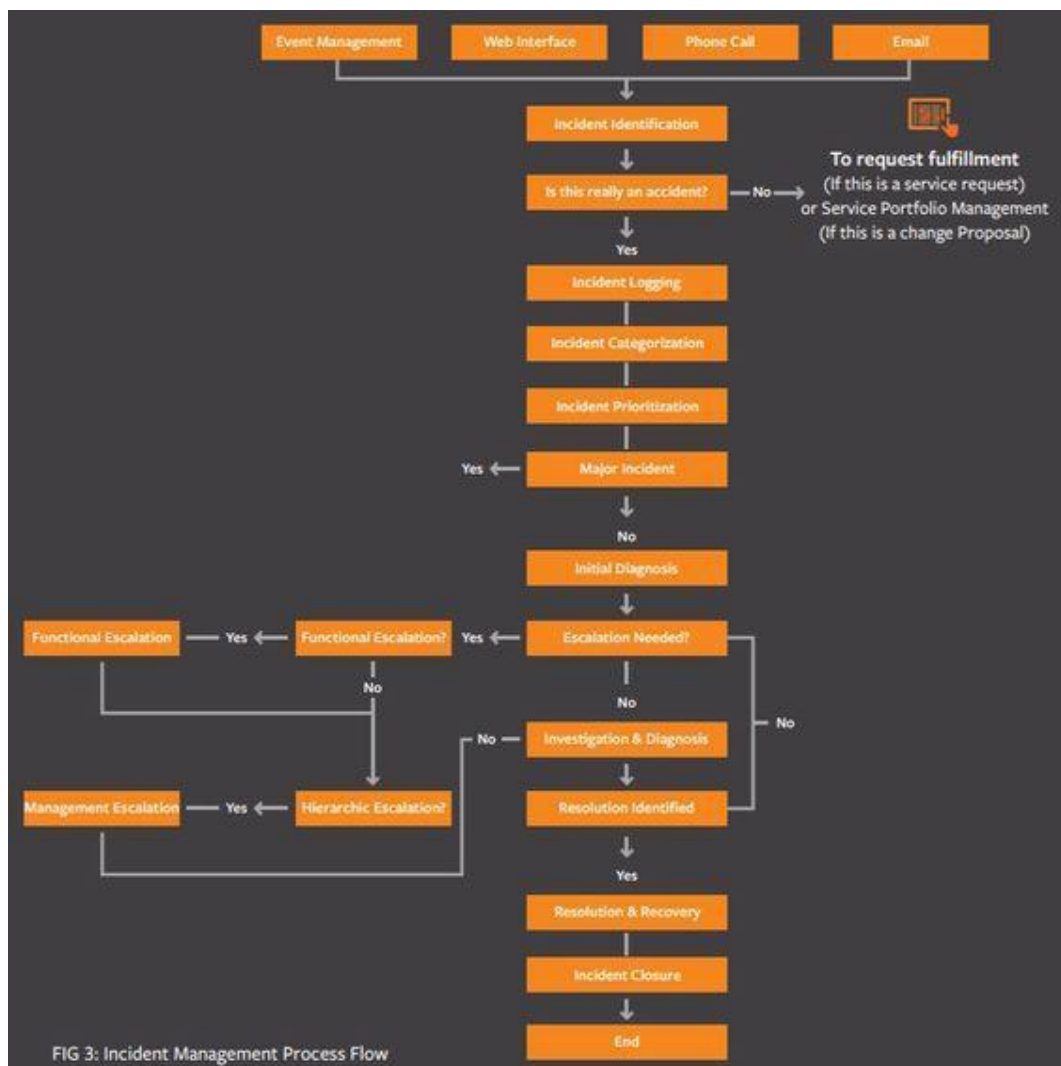


Fig 4.9.1: Data Flow Diagram

CHAPTER 5

RESULTS AND DISCUSSION

In [6]:

```
# There are 258 wbse which dont make sense as it cannot indicate any department So we delete it
# Delete t_Reopen_Time & Category column
DeletColList = ['c_WBS']

# Delete the Listed columns / features
df1.drop(DeletColList, inplace=True, axis=1)

# Index the columns to keep catagorical, continous and outcome separately
df1=df1.sort_index(axis=1)

df1.head()
```

Out[6]:

	c_Cl_Cat	c_Closure_Code	q_No_of_Reassignments	q_No_of_Related_Incidents	q_No_of_Related_Interactions	t_Close_Time	t_Open_Time	t_ReopenFlag	t_
0	11	4	26	2	1	04-11-13 13.51	05-02-12 13.32	0	
1	1	6	33	1	1	02-12-13 12.36	12-03-12 15.44	1	
2	1	3	13	0	1	14-11-13 9.31	17-07-12 11.49	0	
3	1	4	2	0	1	08-11-13 13.55	10-08-12 11.01	0	
4	1	4	4	0	1	08-11-13 13.54	10-08-12 11.27	0	

Fig 5.1: Service request logging

Once identified as an Service request, the service desk logs the Service request as a ticket. The ticket should include information, such as the user's name and contact information, the Service request description, and the date and time of the Service request report (for SLA adherence).The logging process can also include categorization, prioritization, and the steps the service desk completes.

```
In [23]: # Almost All P1/P2 have related incidents in past
rcParams['figure.figsize'] = 20,10
Crosstab_NR = pd.crosstab(df1.q_No_of_Related_Incidents, df1.y_Priority)
#.round(6)*100
print(Crosstab_NR)
Crosstab_NR.plot(kind='barh', stacked=False, grid=True, title="Related Incidents Vs Priority: Calculated as % Priority")
```

y_Priority q_No_of_Related_Incidents	1	2	3	4	5
0	0	345	4836	21509	8079
1	1	54	198	454	220
2	0	8	13	38	8
3	0	5	7	6	1
4	0	2	4	2	3
5	0	1	1	4	1
6	0	1	1	3	3
7	0	0	2	1	0
8	0	1	3	1	0
9	0	3	0	1	2
10	0	0	0	0	1
11	0	1	1	0	1
12	0	2	0	2	0
13	0	0	1	0	0
14	0	0	0	1	0
15	0	1	0	0	0
16	0	1	0	1	1
17	1	1	1	0	0
21	0	1	0	0	0
24	0	2	0	0	0
25	0	0	0	0	1
26	1	0	0	0	0
54	0	1	0	0	0
63	0	1	0	0	0

Fig 5.2 Service request categorization

Categorization involves assigning a category and at least one subcategory to the Service request. This action serves several purposes. First, it allows the service desk to sort and model Service requests based on their categories and subcategories. Second, it allows some issues to be automatically prioritized. For example, an Service request might be categorized as “network” with a sub-category of “network outage”. The third purpose is to provide accurate Service request tracking. When Service requests are categorized, patterns emerge. It’s easy to quantify how often certain Service requests come up and point to trends that require training or problem management. For example, it’s much easier to sell the CFO on new hardware when the data supports the decision.

```
In [14]: # Show the chrn has happened in for account lengths in durations 5 days. Also Majority of P1/P2 have been solved in 5 days
#seem to have been solved in with in 5 days
Crosstab_AL = pd.crosstab(df1.t_TicketWIPDurationDays_bins, df1.y_Priority)
print(Crosstab_AL)
#Crosstab_AL.plot(kind='barh', stacked=False, grid=True, title="Ticket Resolved Duration vs Priority: Calculated as % of overall

y_Priority      1      2      3      4      5
t_TicketWIPDurationDays_bins
(0.0, 5.0]      2    417    4474   17956   5634
(5.0, 10.0]     1     11     288    1952   1460
(10.0, 15.0]    0      2    107     749    539
(15.0, 20.0]    0      0     33     311    207
(20.0, 25.0]    0      1     33     219    161
...            ..    ...    ...    ...    ...
(480.0, 485.0]  0      0      0        1      0
(485.0, 490.0]  0      0      0        1      0
(495.0, 500.0]  0      0      0        1      0
(625.0, 630.0]  0      0      1         0      0
(635.0, 640.0]  0      0      0         1      0

[79 rows x 5 columns]
```

Fig 5.3: Service request prioritization

Service request prioritization is important for SLA response adherence. An Service request's priority is determined by its impact on users and on the business and its urgency. Urgency is how quickly a resolution is required; impact is the measure of the extent of potential damage the Service request may cause. Low-priority Service requests are those that do not interrupt users or the business and can be worked around. Services to users and customers can be maintained. Medium-priority Service requests affect a few staff and interrupt work to some degree. Customers may be slightly affected or inconvenienced. High-priority Service requests affect a large number of users or customers, interrupt business, and affect service delivery. These Service requests almost always have a financial impact.

```
In [21]: # Predicting and converting into Dataframe
#y1_predict = grid_model_XGB.predict(X1_test)
y1_predict = pd.DataFrame(grid_model_XGB.predict(X1_test))
#,columns=['y_Predict_Priority']
Proba = grid_model_XGB.predict_proba(X1_test)
y1_predict.shape
#y1_test.shape
```

Out[21]: (7170, 1)

```
In [22]: # Results
print(accuracy_score(y1_test,y1_predict)*100)
#print(recall_score(y1_test,y1_predict)*100)
#print(classification_report(y1_test,y1_predict))
confusion_matrix(y1_test,y1_predict)
#Counter(y1_predict)
```

83.76569037656904

```
Out[22]: array([[ 29,   18,   32,    4],
 [ 15,  714,  304,   49],
 [ 12,   77, 4064,  191],
 [  1,   47,  414, 1199]], dtype=int64)
```

Fig 5.4: Accuracy Score

CHAPTER 6

CONCLUSION

In this system, Invest in emerging technology to support the goals of incident management, which will lead to real-time incident management capabilities. Speed and performance of incident resolution have always been a priority for many organizations, so they can serve their customers better. To be more effective at this task, incident management capabilities must be moved from the service desk analyst and closer to supporting the customers of the service. Business as usual must resume as soon as possible to minimize disruption and maybe even to save lives. Disruption is not always defined as downtime of a system, but may also be a service incapable of functioning normally. This could indicate performance issues related to accessing and utilizing the service. At this point, the Service request is considered closed and the Service request process ends.

REFERENCE

- [1] DHS, "Cyber incident reporting: a unified message for reporting to the federal government," US Department of Homeland Security, 2016.
- [2] J. Raikes and G. McBean, "Responsibility and liability in emergency management to natural disaster: A Canadian example," *Int. J. Disaster Risk Reduction*, vol. 16, pp. 12–18, 2016.
- [3] C. Onwubiko, "Understanding cyber situation awareness," *Int. J. Cyber Situational Awareness*, vol. 1, no. 1, pp. 11–30, 2016.
- [4] J. Wang, "Anatomy of a security operations center," GFirst 2010, NASA, Washington, DC. Accessed on: Jan. 31, 2018. [Online]. Available:
- [5] NIST, "Guide for cybersecurity event recovery," Special publication 800-184, 2016, doi: 10.6028/NIST.SP.800-184.
- [6] C. Onwubiko, "CoCoa: An ontology for cybersecurity operations centre analysis process," in *Proc. Int. Conf. Cyber Situational Awareness, Data Analytics Assessment*, 2018, pp. 1–8,
- [7] US CERT, "National cyber incident response plan," Homeland Security, 2016. [8] NIST, "Computer security incident handling guide," Special Publication 800-61r2, 2012.
- [9] C. D. McDermott, W. Haynes, and A. V. Petrovski, "Threat detection and analysis in the internet of things using deep packet inspection," *Int. J. Cyber Situational Awareness*, vol. 3, no. 1, pp. 61–83, 2018, doi: 10.22619/IJCSA.2018.100118.
- [10] C. Onwubiko, "Security operations centre: Situation awareness, threat intelligence and cybercrime," in *Proc. Int. Conf. Cyber Situational Awareness, Data Analytics Assessment*, 2017, pp. 1–6.
- [11] NCSC, "Incident management," 2018. [Online]. Available: <https://www.Ncsc.gov.uk/incident-management>

APPENDIX

A. Plagiarism Report

SERVICE REQUEST CLASSIFICATION BASED ON PRIORITY (1).docx

ORIGINALITY REPORT

11 %

SIMILARITY INDEX

5 %

INTERNET SOURCES

4 %

PUBLICATIONS

7 %

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Campbellsville University

Student Paper

5 %

2

Cyril Onwubiko, Karim Ouazzane. "SOTER: A Playbook for Cybersecurity Incident Management", IEEE Transactions on Engineering Management, 2020

Publication

4 %

3

www.jpinfotech.org

Internet Source

2 %

4

repository.londonmet.ac.uk

Internet Source

1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On

B. Source Code

```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {},
      "source": [
        "# Optimize XG Boost - parameters / hyper parameters"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 1,
      "metadata": {},
      "outputs": [],
      "source": [
        "import pandas as pd\n",
        "import numpy as np\n",
        "import researchpy as rp\n",
        "from scipy import stats\n",
        "\n",
        "import matplotlib.pyplot as plt\n",
        "import pydotplus\n",
        "from matplotlib import rcParams\n",
        "import seaborn as sb\n"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 2,
      "metadata": {},
      "outputs": [],
      "source": [
        "# # Find no. of factors in y\n",
        "from collections import Counter\n",
        "# #label encoding\n",
        "from sklearn.preprocessing import LabelEncoder, scale,
MinMaxScaler, StandardScaler,OneHotEncoder,LabelBinarizer\n",
        "#from sklearn.decomposition import PCA"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 3,
      "metadata": {},
      "outputs": [
        {
          "name": "stdout",
          "output_type": "stream",
          "text": [
```

```

    "Requirement already satisfied: xgboost in c:\\users\\w10-dell\\anaconda3\\lib\\site-
packages (1.3.3)\n",
    "Requirement already satisfied: scipy in c:\\users\\w10-
dell\\anaconda3\\lib\\site-packages (from xgboost) (1.4.1)\n",
    "Requirement already satisfied: numpy in c:\\users\\w10-dell\\anaconda3\\lib\\site-
packages (from xgboost) (1.18.1)\n"
  ]
},
],
"source": [
  "# Import All classification Algorithms\n",
  "\n",
  "!pip install xgboost\n",
  "from xgboost import XGBClassifier"
],
},
{
  "cell_type": "code",
  "execution_count": 4,
  "metadata": {},
  "outputs": [],
  "source": [
    "from sklearn.model_selection import train_test_split,
GridSearchCV,cross_val_score,validation_curve,KFold"
  ]
},
{
  "cell_type": "code",
  "execution_count": 5,
  "metadata": {},
  "outputs": [],
  "source": [
    "# Metrics selection\n",
    "from sklearn.metrics import accuracy_score, recall_score,
confusion_matrix,precision_recall_fscore_support,classification_report,roc_curve,auc"
  ]
},
{
  "cell_type": "code",
  "execution_count": 6,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "Index(['c_CI_Cat', 'c_CI_Name', 'c_CI_Subcat', 'c_Closure_Code', 'c_WBS',\n      'q_No_of_Reassignments', 'q_No_of_Related_Incidents',\n      'q_No_of_Related_Interactions', 't_Close_Time', 't_Open_Time',\n      'y_Priority', 't_ReopenFlag', 't_TicketWIPDurationDays'],\n      dtype='object')"
        ]
      }
    ]
  },
  "execution_count": 6,
  "metadata": {},

```

```

        "output_type": "execute_result"
    }
],
    "source": [
        "# Open files Both files are opened as ny drop or changes in feature will be done
is both normal and one hot file\n",
        "df1 = pd.read_csv('3_ITSM_cleansed_Encoded_Engineered_All.csv')\n",
        "df1.columns"
    ]
},
{
    "cell_type": "code",
    "execution_count": 7,
    "metadata": {},
    "outputs": [],
    "source": [
        "CatVarList = ['c_CI_Cat', 'c_CI_Name', 'c_CI_Subcat',
'c_Closure_Code', 'c_WBS', 't_ReopenFlag']\n",
        "ConVarList = ['q_No_of_Reassignments',
'q_No_of_Related_Incidents', 'q_No_of_Related_Interactions', 't_TicketWIPDurationDays'
]\n",
        "DatVarList = ['t_Close_Time', 't_Open_Time']\n",
        "OutValList = ['y_Priority']\n", "\n",

        "X_All_1 = ['c_CI_Cat', 'c_CI_Name', 'c_CI_Subcat',
'c_Closure_Code', 'c_WBS', 't_ReopenFlag',\n",
        "            'q_No_of_Reassignments',
'q_No_of_Related_Incidents', 'q_No_of_Related_Interactions', 't_TicketWIPDurationDays']\n",

        "
    ]
},
{
    "cell_type": "code",
    "execution_count": 8,
    "metadata": {},
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "(35846, 10)\n",
                "(35846,)\n"
            ]
        }
    ],
    "source": [
        "# Assign varibale and X and y\n",
        "X1 = df1[X_All_1]\n",
        "y1 = df1.y_Priority\n",
        "\n",
        "print(X1.shape)\n",
        "print(y1.shape)"
    ]
}
]

```

```

},
{
  "cell_type": "code",
  "execution_count": 9,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "(28676, 10)\n",
        "(28676,)\n",
        "(7170, 10)\n",
        "(7170,)\n"
      ]
    }
  ],
  "source": [
    "# Split 80:20\n",
    "split_seed = 10\n",
    "X1_train, X1_test, y1_train, y1_test =\n",
    "train_test_split(X1,y1,random_state=split_seed, test_size=.2)\n",
    "print(X1_train.shape)\n",
    "print(y1_train.shape)\n",
    "print(X1_test.shape)\n",
    "print(y1_test.shape)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 10,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "23572  4\n",
          "163    3\n",
          "30420  3\n",
          "21679  3\n",
          "23989  3\n",
          "Name: y_Priority, dtype: int64"
        ]
      }
    }
  ],
  "execution_count": 10,
  "metadata": {},
  "output_type": "execute_result"
}
],
"source": [
  "y1_train.head()"
]
},
{

```

```

"cell_type": "code",
"execution_count": 11,
"metadata": {},
"outputs": [
  {
    "data": {
      "text/plain": [
        "Counter({4: 6660, 3: 17679, 2: 3986, 1: 348, 0: 3})"
      ]
    },
    "execution_count": 11,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "Counter(y1_train)"
]
},
{
  "cell_type": "markdown",
  "metadata": {},
  "source": [
    "## Scaling done based on EDA analysis. Fit on the Train set, but tranform both transform both training and test set"
  ]
},
{
  "cell_type": "code",
  "execution_count": 12,
  "metadata": {},
  "outputs": [
    {
      "name": "stderr",
      "output_type": "stream",
      "text": [
        "C:\\Users\\W10-Dell\\anaconda3\\lib\\site-packages\\ipykernel_launcher.py:6: SettingWithCopyWarning: \\n",
        "A value is trying to be set on a copy of a slice from a DataFrame.\\n",
        "Try using .loc[row_indexer,col_indexer] = value instead\\n", "\\n",
        "\\n",
        "See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy\\n",
        "C:\\Users\\W10-Dell\\anaconda3\\lib\\site-packages\\pandas\\core\\indexing.py:965: SettingWithCopyWarning: \\n",
        "A value is trying to be set on a copy of a slice from a DataFrame.\\n",
        "Try using .loc[row_indexer,col_indexer] = value instead\\n", "\\n",
        "\\n",
        "See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy\\n",
        "self.obj[item] = s\\n",
        "C:\\Users\\W10-Dell\\anaconda3\\lib\\site-packages\\ipykernel_launcher.py:7: SettingWithCopyWarning: \\n",

```

```

"A value is trying to be set on a copy of a slice from a DataFrame.\n",
"Try using .loc[row_indexer,col_indexer] = value instead\n", "\n",

"See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy\n",
" import sys\n",
"C:\\Users\\W10-Dell\\anaconda3\\lib\\site-packages\\pandas\\core\\indexing.py:965:
SettingWithCopyWarning: \n",
"A value is trying to be set on a copy of a slice from a DataFrame.\n",
"Try using .loc[row_indexer,col_indexer] = value instead\n", "\n",

"See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy\n",
" self.obj[item] = s\n"
]
}
],
"source": [
"# Scale the data\n",
"scaler = StandardScaler()\n",
"# Fit on train data\n",
"scaler.fit(X1_train[ConVarList])\n",
"# Transform both train and test\n",
"X1_train[ConVarList] = scaler.transform(X1_train[ConVarList])\n",
"X1_test[ConVarList] = scaler.transform(X1_test[ConVarList])"
],
{
"cell_type": "code",
"execution_count": 13,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"(28676, 10)\n",
"(7170, 10)\n"
]
}
],
"source": [
"print(X1_train.shape)\n",
"print(X1_test.shape)"
],
{
"cell_type": "markdown",
"metadata": {},
"source": [
"### Grid Search to optimize the hyper parameters"
]
},
{

```

```

"cell_type": "code",
"execution_count": 14,
"metadata": {},
"outputs": [
  {
    "name": "stderr",
    "output_type": "stream",
    "text": [
      "C:\\Users\\W10-Dell\\anaconda3\\lib\\site-
packages\\sklearn\\model_selection\\_split.py:667: UserWarning: The least
populated class in y has only 3 members, which is less than n_splits=5.\\n",
      " % (min_groups, self.n_splits)), UserWarning)\\n", "C:\\Users\\W10-
Dell\\anaconda3\\lib\\site-packages\\xgboost\\sklearn.py:888:
UserWarning: The use of label encoder in XGBClassifier is deprecated and will be
removed in a future release. To remove this warning, do the following: 1) Pass option
use_label_encoder=False when constructing XGBClassifier object; and 2) Encode
your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].\\n",
      " warnings.warn(label_encoder_deprecation_msg, UserWarning)\\n"
    ]
  },
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "[21:28:49] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \\n",
      "Parameters: { scoring } might not be used.\\n",
      "\\n",
      " This may not be accurate due to some parameters are only used in language
bindings but\\n",
      " passed down to XGBoost core. Or some parameters are not used but slip through
this\\n",
      " verification. Please open an issue if you find above cases.\\n",
      "\\n",
      "\\n",
      "[21:28:49] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\\n",
      "[21:28:55] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \\n",
      "Parameters: { scoring } might not be used.\\n",
      "\\n",
      " This may not be accurate due to some parameters are only used in
language bindings but\\n",
      " passed down to XGBoost core. Or some parameters are not used but slip through
this\\n",
      " verification. Please open an issue if you find above cases.\\n",
      "\\n",
      "\\n",
      "[21:28:56] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\\n",

```

```

"[21:29:07] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",
  "Parameters: { scoring } might not be used.\n",
  "\n",
  " This may not be accurate due to some parameters are only used in
language bindings but\n",
  " passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
  " verification. Please open an issue if you find above cases.\n",
  "\n",
  "\n",
  "\n",
"[21:29:07] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n",
"[21:29:19] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",
  "Parameters: { scoring } might not be used.\n",
  "\n",
  " This may not be accurate due to some parameters are only used in
language bindings but\n",
  " passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
  " verification. Please open an issue if you find above cases.\n",
  "\n",
  "\n",
  "\n",
"[21:29:19] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n",
"[21:29:30] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",
  "Parameters: { scoring } might not be used.\n",
  "\n",
  " This may not be accurate due to some parameters are only used in
language bindings but\n",
  " passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
  " verification. Please open an issue if you find above cases.\n",
  "\n",
  "\n",
  "\n",
"[21:29:30] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n",
"[21:29:42] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",
  "Parameters: { scoring } might not be used.\n",
  "\n",
  " This may not be accurate due to some parameters are only used in
language bindings but\n",
  " passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
  " verification. Please open an issue if you find above cases.\n",

```

```

"\n",
"\n",
"[21:29:42] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n"
]
},
{
  "data": {
    "text/plain": [
      "GridSearchCV(cv=None, error_score=nan,\n",
      "      estimator=XGBClassifier(base_score=None, booster='gbtree',\n",
      "      colsample_bylevel=None,\n",
      "      colsample_bynode=None,\n",
      "      colsample_bytree=None, gamma=None,\n",
      "      gpu_id=None, importance_type='gain',\n",
      "      interaction_constraints=None,\n",
      "      learning_rate=None, max_delta_step=None,\n",
      "      max_depth=None, min_child_weight=None,\n",
      "      missing=nan, monotone_constraints=None,...\n",
      "      reg_lambda=None, scale_pos_weight=None,\n",
      "      subsample=None, tree_method=None,\n",
      "      use_label_encoder=True,\n",
      "      validate_parameters=None, verbosity=None),\n",
      "      iid='deprecated', n_jobs=None,\n",
      "      param_grid={'colsample_bytree': [0.7], 'gamma': [0.1],\n",
      "      'learning_rate': [0.05], 'max_depth': [15],\n",
      "      'min_child_weight': [5], 'scoring': ['recall']},\n",
      "      pre_dispatch='2*n_jobs', refit=True, return_train_score=False,\n",
      "      scoring=None, verbose=0)"
    ]
  },
  "execution_count": 14,
  "metadata": {},
  "output_type": "execute_result"
}
],
"source": [
  "# Parameters used for Grid search to get best parameters. Use this again only to
  get best parameters..its take a lot of time.\n",
  """\n",
  "parameters = {\n",
  "    \"learning_rate\": [.05,.1, .15],\n",
  "    \"max_depth\": [3,5,8,10,15,20],\n",
  "    \"min_child_weight\" : [ 3, 5, 7 ],\n",
  "    \"gamma\" : [ 0.0, 0.1],\n",
  "    \"colsample_bytree\" : [ 0.1,0.7,0.8],\n",
  "    \"scoring\" : [\"recall\"]\n",
  "  }\n",
  """\n",
  "parameters = {\n",
  "    \"learning_rate\": [.05],\n",
  "    \"max_depth\": [15],\n",
  "    \"min_child_weight\" : [5],\n",
  "    \"gamma\": [0.1],

```

```

"        \"colsample_bytree\" : [.7], \"n\",
"        \"scoring\"       : [\"recall\"] \"n\",
"    }\"n\",
\"n\",
"model = XGBClassifier(booster='gbtree',random_state=380)\"n\",
"grid_model_XGB = GridSearchCV(model, parameters)\"n\",
"grid_model_XGB.fit(X1_train, y1_train)\"
]
},
{
"cell_type": "code",
"execution_count": 15,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"{'colsample_bytree': 0.7,\"n\",
" 'gamma': 0.1,\"n\",
" 'learning_rate': 0.05,\"n\",
" 'max_depth': 15,\"n\",
" 'min_child_weight': 5,\"n\",
" 'scoring': 'recall'}\"
]
},
"execution_count": 15,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"grid_model_XGB.best_params_\"
]
},
{
"cell_type": "code",
"execution_count": 16,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"83.75\"n\",
"        precision  recall  f1-score  support\"n\",
\"n\",
"         1    0.53    0.35    0.42      83\"n\",
"         2    0.83    0.66    0.73    1082\"n\",
"         3    0.84    0.94    0.89   4344\"n\",
"         4    0.83    0.72    0.77   1661\"n\",
\"n\",
" accuracy                0.84    7170\"n\",
" macro avg          0.76    0.67    0.707170\"n\",
" weighted avg       0.84    0.84    0.837170\"n\",

```

```

        "\n"
    ]
}
],
"source": [
    "y1_predict =
pd.DataFrame(grid_model_XGB.predict(X1_test),columns=['y_Priority'])\n",
    "print(accuracy_score(y1_test,y1_predict).round(4)*100)\n",
    "#print(recall_score(y1_test,y1_predict).round(4)*100)\n",
    "print(classification_report(y1_test,y1_predict))"
]
},
{
    "cell_type": "code",
    "execution_count": 17,
    "metadata": {},
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "(7170,)\n",
                "(7170, 1)\n"
            ]
        }
    ],
    "source": [
        "print(y1_test.shape)\n",
        "print(y1_predict.shape)"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {},
    "source": [
        "## Vary Model Random state to see if the results can be further improved"
    ]
},
{
    "cell_type": "code",
    "execution_count": 18,
    "metadata": {},
    "outputs": [
        {
            "data": {
                "text/plain": [
                    ""\nparameters = {\nlearning_rate\": [.05],\nmax_depth\": [15],\nmin_child_weight\": [5],\ngamma\": [0.1],\ncolsample_bytree\": [.7],\nscore\": [recall]\n}\n\nstore = []\nstore_max=0\nfor i in range(10,500,5):\n    model =\n    XGBClassifier(booster='gbtree',random_state=i)\n    grid_model_XGB =\n    GridSearchCV(model, parameters)\n    grid_model_XGB.fit(X1_train, y1_train)\n    y1_predict = grid_model_XGB.predict(X1_test)\n    if(accuracy_score(y1_test,y1_predict)>=store_max):\n"
                ]
            },
        }
    ]
}

```

```

store.append([i,accuracy_score(y1_test,y1_predict)*100])\n
store_max=accuracy_score(y1_test,y1_predict)\n
print(i,accuracy_score(y1_test,y1_predict)*100)\n"
    ]
    },
    "execution_count": 18,
    "metadata": {},
    "output_type": "execute_result"
}
],
"source": [
"# masked after Running\n",
"""\n",
"parameters = {\nlearning_rate\n": [.05],\n",
"          \nmax_depth\n": [15],\n",
"          \nmin_child_weight\n": [5],\n",
"          \ngamma\n": [0.1],\n",
"          \ncolsample_bytree\n": [.7], \n",
"          \nscore\n": [\nrecall\n"] \n",
"          }\n",
"\n",
"store = []\n",
"store_max=0\n",
"for i in range(10,500,5):\n",
"    model = XGBClassifier(booster='gbtree',random_state=i)\n",
"    grid_model_XGB = GridSearchCV(model, parameters)\n",
"    grid_model_XGB.fit(X1_train, y1_train)\n",
"    y1_predict = grid_model_XGB.predict(X1_test)\n",
"    if(accuracy_score(y1_test,y1_predict)>=store_max):\n",
"        store.append([i,accuracy_score(y1_test,y1_predict)*100])\n",
"        store_max=accuracy_score(y1_test,y1_predict)\n",
"    print(i,accuracy_score(y1_test,y1_predict)*100)\n",
"""\n",
"# maxmim we find at random_state =90 @ 83.9"
]
},
{
"cell_type": "code",
"execution_count": 19,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"Stored variables and their in-db values:\n"
]
}
],
"source": [
"store"
]
},
{

```

```

"cell_type": "markdown",
"metadata": {},
"source": [
    "## Final Model"
]
},
{
    "cell_type": "code",
    "execution_count": 20,
    "metadata": {},
    "outputs": [
        {
            "name": "stderr",
            "output_type": "stream",
            "text": [
                "C:\\Users\\W10-Dell\\anaconda3\\lib\\site-
packages\\sklearn\\model_selection\\_split.py:667: UserWarning: The least
populated class in y has only 3 members, which is less than n_splits=5.\\n",
                " % (min_groups, self.n_splits)), UserWarning)\\n", "C:\\Users\\W10-
Dell\\anaconda3\\lib\\site-packages\\xgboost\\sklearn.py:888:
UserWarning: The use of label encoder in XGBClassifier is deprecated and will be
removed in a future release. To remove this warning, do the following: 1) Pass option
use_label_encoder=False when constructing XGBClassifier object; and 2) Encode
your labels (y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].\\n",
                " warnings.warn(label_encoder_deprecation_msg, UserWarning)\\n"
            ]
        },
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "[21:30:56] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \\n",
                "Parameters: { scoring } might not be used.\\n",
                "\\n",
                " This may not be accurate due to some parameters are only used in
language bindings but\\n",
                " passed down to XGBoost core. Or some parameters are not used but slip through
this\\n",
                " verification. Please open an issue if you find above cases.\\n",
                "\\n",
                "\\n",
                "[21:30:56] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\\n",
                "[21:31:07] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \\n",
                "Parameters: { scoring } might not be used.\\n",
                "\\n",
                " This may not be accurate due to some parameters are only used in language
bindings but\\n",
                " passed down to XGBoost core. Or some parameters are not used but slip through
this\\n",

```

```

" verification. Please open an issue if you find above cases.\n",
"\n",
"\n",
"[21:31:07] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror'
to 'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n",
"[21:31:18] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",
"Parameters: { scoring } might not be used.\n",
"\n",
" This may not be accurate due to some parameters are only used in language
bindings but\n",
" passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
" verification. Please open an issue if you find above cases.\n",
"\n",
"\n",
"[21:31:18] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n",
"[21:31:29] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",
"Parameters: { scoring } might not be used.\n",
"\n",
" This may not be accurate due to some parameters are only used in
language bindings but\n",
" passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
" verification. Please open an issue if you find above cases.\n",
"\n",
"\n",
"[21:31:29] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n",
"[21:31:40] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",
"Parameters: { scoring } might not be used.\n",
"\n",
" This may not be accurate due to some parameters are only used in
language bindings but\n",
" passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
" verification. Please open an issue if you find above cases.\n",
"\n",
"\n",
"[21:31:40] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n",
"[21:31:51] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:541: \n",

```

```

    "Parameters: { scoring } might not be used.\n",
    "\n",
    " This may not be accurate due to some parameters are only used in
language bindings but\n",
    " passed down to XGBoost core. Or some parameters are not used but slip through
this\n",
    " verification. Please open an issue if you find above cases.\n",
    "\n",
    "\n",
    "[21:31:51] WARNING: C:/Users/Administrator/workspace/xgboost-
win64_release_1.3.0/src/learner.cc:1061: Starting in XGBoost 1.3.0, the default
evaluation metric used with the objective 'multi:softprob' was changed from 'merror' to
'mlogloss'. Explicitly set eval_metric if you'd like to restore the old behavior.\n"
]
},
{
  "data": {
    "text/plain": [
      "GridSearchCV(cv=None, error_score=nan,\n",
      "      estimator=XGBClassifier(base_score=None, booster='gbtree',\n",
      "      colsample_bylevel=None,\n",
      "      colsample_bynode=None,\n",
      "      colsample_bytree=None, gamma=None,\n",
      "      gpu_id=None, importance_type='gain',\n",
      "      interaction_constraints=None,\n",
      "      learning_rate=None, max_delta_step=None,\n",
      "      max_depth=None, min_child_weight=None,\n",
      "      missing=nan, monotone_constraints=None,...\n",
      "      reg_lambda=None, scale_pos_weight=None,\n",
      "      subsample=None, tree_method=None,\n",
      "      use_label_encoder=True,\n",
      "      validate_parameters=None, verbosity=None),\n",
      "      iid='deprecated', n_jobs=None,\n",
      "      param_grid={'colsample_bytree': [0.7], 'gamma': [0.1],\n",
      "      'learning_rate': [0.05], 'max_depth': [15],\n",
      "      'min_child_weight': [5], 'scoring': ['recall']},\n",
      "      pre_dispatch='2*n_jobs', refit=True, return_train_score=False,\n",
      "      scoring=None, verbose=0)"
    ]
  },
  "execution_count": 20,
  "metadata": {},
  "output_type": "execute_result"
}
],
"source": [
  "parameters = {\nlearning_rate\n: [.05],\n",
  "      \nmax_depth\n: [15],\n",
  "      \nmin_child_weight\n: [5],\n",
  "      \ngamma\n: [0.1],\n",
  "      \ncolsample_bytree\n: [.7], \n",
  "      \nscoring\n: [\n'recall\n']\n",
  "      }\n",
  "model = XGBClassifier(booster='gbtree',random_state=90)\n",

```

```

"grid_model_XGB = GridSearchCV(model,
parameters)\n", "grid_model_XGB.fit(X1_train, y1_train)"
]
},
{
"cell_type": "code",
"execution_count": 21,
"metadata": {},
"outputs": [
{
"data": {
"text/plain": [
"(7170, 1)"
]
},
"execution_count": 21,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
"# Predicting and converting into Dataframe\n",
"#y1_predict = grid_model_XGB.predict(X1_test)\n",
"y1_predict = pd.DataFrame(grid_model_XGB.predict(X1_test))\n",
"#,columns=['y_Predict_Priority']\n",
"Proba = grid_model_XGB.predict_proba(X1_test)\n",
"y1_predict.shape\n",
"#y1_test.shape"
]
},
{
"cell_type": "code",
"execution_count": 22,
"metadata": {},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"83.76569037656904\n"
]
},
{
"data": {
"text/plain": [
"array([[ 29, 18, 32,   4],\n"
"       [ 15, 714, 304, 49],\n"
"       [ 12,  77, 4064, 191],\n"
"       [  1, 47, 414, 1199]], dtype=int64)"
]
},
"execution_count": 22,
"metadata": {},
"output_type": "execute_result"
}
]

```

```

    }
  ],
  "source": [
    "# Results\n",
    "print(accuracy_score(y1_test,y1_predict)*100)\n",
    "#print(recall_score(y1_test,y1_predict)*100)\n",
    "#print(classification_report(y1_test,y1_predict))\n",
    "confusion_matrix(y1_test,y1_predict)\n",
    "#Counter(y1_predict)"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {},
  "outputs": [],
  "source": []
}
],
"metadata": {
  "kernel_spec": {
    "display_name": "Python 3",
    "language": "python",
    "name": "python3"
  },
  "language_info": {
    "codemirror_mode": {
      "name": "ipython",
      "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",

```

```
"nbconvert_exporter": "python",  
"pygments_lexer": "ipython3",  
"version": "3.7.6"  
}  
,  
"nbformat": 4,  
"nbformat_minor": 2  
}
```