

A NOVEL APPROACH ON ASSOCIATED DIFFERENTIAL SECRECY: FEATURE SELECTION IN MACHINE LEARNING

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

by

KESAVAN (37110341)



**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 **KESAVAN (37110341)** who carried out the project entitled “**A NOVEL APPROACH ON ASSOCIATED DIFFERENTIAL SECRECY: FEATURE SELECTION IN MACHINE LEARNING**” under my supervision from August 2020 to March 2021.

Internal guide

Mrs.Brumancia , M.E.,(Ph.D)

Head of the Department

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KESAVAN** hereby declare that the Project Report entitled “**A NOVEL APPROACH ON ASSOCIATED DIFFERENTIAL SECRECY: FEATURE SELECTION IN MACHINE LEARNING**” is done by me under the guidance of **MS. E.Brumancia , M.E.,(Ph.D)** Department of Information Technology 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 **MS. E.Brumancia ,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

Privacy preserving in machine learning is a crucial issue in industry informatics since data used for training in industries usually contain sensitive information. Existing differentially private machine learning algorithms have not considered the impact of data correlation, which may lead to more privacy leakage than expected in industrial applications. For example, data collected for traffic monitoring may contain some correlated records due to temporal correlation or user correlation. To fill this gap, we propose a correlation reduction scheme with differentially private feature selection considering the issue of privacy loss when data have correlation in machine learning tasks. The proposed scheme involves five steps with the goal of managing the extent of data correlation, preserving the privacy, and supporting accuracy in the prediction results. In this way, the impact of data correlation is relieved with the proposed feature selection scheme, and moreover the privacy issue of data correlation in learning is guaranteed. The proposed method can be widely used in machine learning algorithms which provide services in industrial areas. Experiments show that the proposed scheme can produce better prediction results with machine learning tasks and fewer mean square errors for data queries compared to existing schemes.

INDEX

S.No.	TOPIC	PAGE No.
1.	INTRODUCTION	1-2
	1.1 Scope	1
	1.2 Objective	2
	1.3 Purpose	2
	1.4 Keywords	2
2.	Objective of the Project	3
	A. Existing System & Dis. Advantages	3
	B. Proposed System & Advantages	3
3.	Project Architecture	4
4.	Literature Survey	5
5.	System Analysis	6
	Feasibility Report	6
	a. Technical Feasibility	6
	b. Operational Feasibility	6
	c. Economical Feasibility	7
6.	System Requirement Specification	8-11
	a. Functional Requirements	9
	a.a. Modules	9
	a.b. Software Requirements	9
	a.c. Hardware Requirements	9
	b. Non Functional Requirements	9
	c. SDLC Methodologies	10
	d. System Architecture	11
7.	System Design	12-16
	a. UML Diagrams	
8.	Technology Description & Implementation	17-34
9.	Coding	35-37
10.	Testing & Test Cases	38-41

11.	Input & Output Design	42-43
12.	Output Screens (Forms&Reports)	44
13.	Future Enhancements	45
14.	Conclusion	46
15.	Bibliography	47
16.	Reference Papers	48-49

1. INTRODUCTION

Currently, machine learning becomes an indispensable tool to provide services for human beings in industrial applications, such as Internet of Things (IoT) and smart cities. One main data source used for machine learning in industry is from human's activities. For example, human's data are often collected via smart phones and these data are analyzed to provide some services in smart cities, such as traffic monitoring and smart health. Data collected from human usually contain some sensitive information, as the location information and health data in above examples. When these data are used for machine learning, individual privacy can be leaked. As a popular technique for privacy preserving, differential privacy was first proposed by Dwork et al. Since then, differential privacy has attracted considerable attention because it provides a rigorous mathematical framework for preserving privacy. Recently, differential privacy is widely used to protect the privacy in industrial informatics, such as location privacy protection, smart grids and multi-agent systems. Much work has addressed the privacy issue in machine learning with differential privacy. Chaudhuri provided an output perturbation where the model was trained and then the noise was added to the output and objective perturbation mechanism where a carefully designed linear perturbation item was added to the original loss function derived differentially private stochastic gradient descent mechanisms and tested them empirically in logistic regression. proposed a differentially private deep learning algorithms which was based on a differentially private version of stochastic gradient descent. studied the differentially private publishing model. However, previous works have not considered the data correlation when designing differentially private machine learning algorithms.

1.1. SCOPE

Machine learning becomes an indispensable tool to provide services for human beings in industrial applications, such as Internet of Things (IoT) and smart cities. One main data source used for machine learning in industry is from human's activities. For example, human's data are often collected via smart phones and these data are analyzed to provide some services in smart cities. Data collected from human usually contain some sensitive information, as the location information and health data in above examples. When these data are used for machine learning, individual privacy can be leaked. So differential privacy has attracted considerable attention because it provides a rigorous mathematical framework for preserving privacy. Recently,

differential privacy is widely used to protect the privacy in industrial informatics, such as location privacy protection, smart grids and multi-agent systems. Hence to stop the more privacy leakage from data we've proposed the Correlated differential privacy to stop the data leakage due to the correlated data.

1.2 OBJECTIVE

Privacy preserving in machine learning is a crucial issue in industry informatics since data used for training in industries usually contain sensitive information. Existing differentially private machine learning algorithms have not considered the impact of data correlation, which may lead to more privacy leakage than expected in industrial applications. So we proposed correlation reduction scheme with differentially private feature selection considering the issue of privacy loss when data have correlation in machine learning tasks.. In this way, the impact of data correlation is relieved with the proposed feature selection scheme, and moreover the privacy issue of data correlation in learning is guaranteed. The proposed method can be widely used in machine learning algorithms which provide services in industrial areas.

1.3 PURPOSE OF PROJECT

One main data source used for machine learning in industry is from human's activities. For example, human's data are often collected via smart phones and these data are analyzed to provide some services in smart cities. Data collected from human usually contain some sensitive information, as the location information and health data in above examples. When these data are used for machine learning, individual privacy can be leaked. So differential privacy has attracted considerable attention because it provides a rigorous mathematical framework for preserving privacy. Recently, differential privacy is widely used to protect the privacy in industrial informatics, such as location privacy protection, smart grids and multi-agent systems. Hence to stop the more privacy leakage from data we've proposed the Correlated differential privacy to stop the data leakage due to the correlated data.

1.4 KEYWORDS

Differential privacy, machine learning, data correlation, feature selection

2. OBJECTIVE OF THE PROJECT

A. Existing System:

Existing differentially private machine learning algorithms have not considered the impact of data correlation, which may lead to more privacy leakage than expected in industrial applications. Differential privacy has attracted considerable attention because it provides a rigorous mathematical framework for preserving privacy. Recently, differential privacy is widely used to protect the privacy in industrial informatics, such as location privacy protection , smart grids and multi-agent systems.

Disadvantages

- It does not considered the impact of data correlation.
- It may lead to more privacy leakage than expected in industrial applications

B. Proposed system:

we propose a correlation reduction scheme with differentially private feature selection considering the issue of privacy loss when data have correlation in machine learning tasks. The proposed scheme involves steps with the goal of managing the extent of data correlation, preserving the privacy, and supporting accuracy in the prediction results. In this way, the impact of data correlation is relieved with the proposed feature selection scheme, and moreover the privacy issue of data correlation in learning is guaranteed. The proposed method can be widely used in machine learning algorithms which provide services in industrial areas. Experiments show that the proposed scheme can produce better prediction results with machine learning tasks and fewer mean square errors for data queries compared to existing schemes.

Advantages:

- The privacy issue of data correlation in learning is guaranteed.
- The proposed method can be widely used in machine learning algorithms which provide services in industrial areas.

C. Modules Description

Supervised Regression (Training Dataset)

The data has been divided into two parts i.e., training and testing data in the 70:30 ratios. Learning algorithms have been applied on the training data and based on the learning, predictions are made on the test data set.

Supervised Regression (Test Dataset)

The test dataset is 30% of the total data. Supervised learning algorithms have been applied on the test data and the output obtained is compared with the actual output.

Pandas: pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.

NumPy: NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.

MatPlotLib: matplotlib. Pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits

Scikit-learn: Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy

LITERATURE SURVEY

4.Literature survey

In the research of location privacy protection, the existing methods are mostly based on the traditional anonymization, fuzzy and cryptography technology, and little success in the big data environment, for example, the sensor networks contain sensitive information, which is compulsory to be appropriately protected. Current trends such as "Industry 4.0" and Internet of Things (IoT), generate, process, and exchange vast amounts of security-critical and privacy-sensitive data, which makes them attractive targets of attacks. However, previous methods overlooked the privacy protection issue, leading to privacy violation. In this paper, we propose a location privacy protection method that satisfying differential privacy constraint to protect location data privacy and maximize the utility of data and algorithm in Industrial Internet of Things. In view of the high value and low density of location data, we combine the utility with the privacy and build a multilevel location information tree model. Furthermore, the index mechanism of differential privacy is used to select data according to the tree node accessing frequency. Finally, the Laplace scheme is used to add noises to accessing frequency of the selecting data. As is shown in the theoretical analysis and the experimental results, the proposed strategy can achieve significant improvements in terms of security, privacy, and applicability.

SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

• FEASIBILITY REPORT

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Economical Feasibility
- Operation Feasibility

A. TECHNICAL FEASIBILITY:

In the feasibility study first step is that the organization or company has to decide that what technologies are suitable to develop by considering existing system.

Here in this application used the technologies like Visual Studio 2008 and SQL Server 2005.

These are free software that would be downloaded from web.

Visual Studio 2008 –it is tool or technology.

B. OPERATIONAL FEASIBILITY:

Not only must an application make economic and technical sense, it must also make operational sense.

Issues to consider when determining the operational feasibility of a project.

Operations Issues	Support Issues
-------------------	----------------

What tools are needed to support operations?	What documentation will users be given?
What skills will operators need to be trained in?	What training will users be given?
What processes need to be created and/or updated?	How will change requests be managed?
What documentation does operations need?	

Very often you will need to improve the existing operations, maintenance, and support infrastructure to support the operation of the new application that you intend to develop. To determine what the impact will be you will need to understand both the current operations and support infrastructure of your organization and the operations and support characteristics of your new application.

To operate this application this system that the user no needs to require any technical knowledge that we are used to develop this project is. Asp.net, C#.net. that the application providing rich user interface by user can do the operation in flexible manner.

C. ECONOMIC FEASIBILITY:

It refers to the benefits or Outcomes we are deriving from the product as compared to the total cost we are spending for developing the product. If the benefits are more or less the same as the older system, then it is not feasible to develop the product.

In the present system, the development of new product greatly enhances the accuracy of the system and cuts short the delay in the processing this application. The errors can be greatly reduced and at the same time providing a great level of security. Here we don't need any additional equipment except memory of required capacity.

No need for spending money on client for maintenance because the database used is web enabled database.

SYSTEM REQUIREMENT SPECIFICATIONS

4.SYSTEM REQUIREMENT SPECIFICATIONS

A **Software Requirements Specification (SRS)** – a requirements specification for a software system – is a complete description of the behavior of a system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A business analyst, sometimes titled system analyst, is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the systems development life cycle domain, typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

- **Business requirements** describe in business terms *what* must be delivered or accomplished to provide value.
- **Product requirements** describe properties of a system or product (which could be one of Several ways to accomplish a set of business requirements.)
- **Process requirements** describe activities performed by the developing organization. For instance, process requirements could specify specific methodologies that must be followed, and constraints that the organization must obey.

Product and process requirements are closely linked. Process requirements often specify the activities that will be performed to satisfy a product requirement. For example, a maximum development cost requirement (a process requirement) may be imposed to help achieve a maximum sales price requirement (a product requirement); a requirement that the product be maintainable (a Product requirement) often is addressed by imposing requirements to follow particular development styles.

A. Functional Requirement

a. Software Requirements

OS : Windows

Python IDE : python 3.x and above

Jupyter Notebook,

Anaconda 3.5

Setup tools and pip to be installed for 3.6.x and above

b. Hardware Requirements

RAM : 4GB and Higher

Processor : Intel i3 and above

Hard Disk : 500GB: Minimum

B. NON FUNCTIONAL REQUIREMENTS

- Secure access of confidential data (user's details). SSL can be used.
- 24 X 7 availability.
- Better component design to get better performance at peak time
 - Flexible service based architecture will be highly desirable for future extension

C. SDLC Methodologies

SDLC MODEL:



The Software Development Lifecycle (SDLC) for small to medium database application development efforts.

This project uses iterative development lifecycle, where components of the application are developed through a series of tight iteration. The first iteration focus on very basic functionality, with subsequent iterations adding new functionality to the previous work and or correcting errors identified for the components in production.

The six stages of the SDLC are designed to build on one another, taking outputs from the previous stage, adding additional effort, and producing results that leverage the previous effort and are directly traceable to the previous stages. During each stage, additional information is gathered or developed, combined with the inputs, and used to produce the stage deliverables. It is important to not that the additional information is restricted in scope, new ideas that would take the project in directions not anticipated by the initial set of high-level requirements or features that are out-of-scope are preserved for later consideration.

Too many software development efforts go awry when development team and customer personnel get caught up in the possibilities of automation. Instead of focusing on high priority features, the team can become mired in a sea of nice to have features that are not essential to solve the problem, but in themselves are highly attractive. This is the root cause of large percentage of failed and or abandoned development efforts and is the primary reason the development team utilizes the iterative model.

Roles and Responsibilities of PDR AND PER

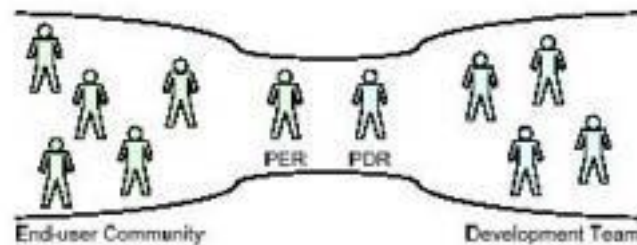
The iterative lifecycle specifies two critical roles that act together to clearly communicate project issues and concepts between the end-user community and the development team.

Primary End-user Representative (PER)

The PER is a person who acts as the primary point of contact and principal approver for the end-user community. The PER is also responsible for ensuring that appropriate subject matter experts conduct end-user reviews in a timely manner.

PER-PDR Relationship

The PER and PDR are the brain trust for the development effort. The PER has the skills and domain knowledge necessary to understand the issues associated with the business processes to be supported by the application and has a close working relationship with the other members of the end-user community. The PDR has the same advantages regarding the application development process and the other members of the development team together, they act as the concentration points for knowledge about the application to be developed.



The objective of this approach is to create the close relationship that is characteristic of a software project with one developer and one end-user in essence, this approach the “pair programming” concept from Agile methodologies and extends it to the end-user community. While it is difficult to create close relationships between the diverse members of an end-user community and a software development team, it is much simpler to create a close relationship between the lead representatives for each group.

When multiple end-users are placed into relationship with multiple members of a development team, communication between the two groups degrades as the number of participants grows. In this model, members of end-user community may communicate with members of the development team as needed, but it is the responsibility of all participants to keep the PER and PDR apprised of the communications for example, this allows the PER and PDR to resolve conflicts that arise when two different end-users communicate different requirements for the same application feature to different members of the development team.

SYSTEM DESIGN

5. SYSTEM DESIGN

a.UML DIAGRAMS

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- actors
- business processes
- (logical) components
- activities
- programming language statements
- database schemas, and
- Reusable software components.

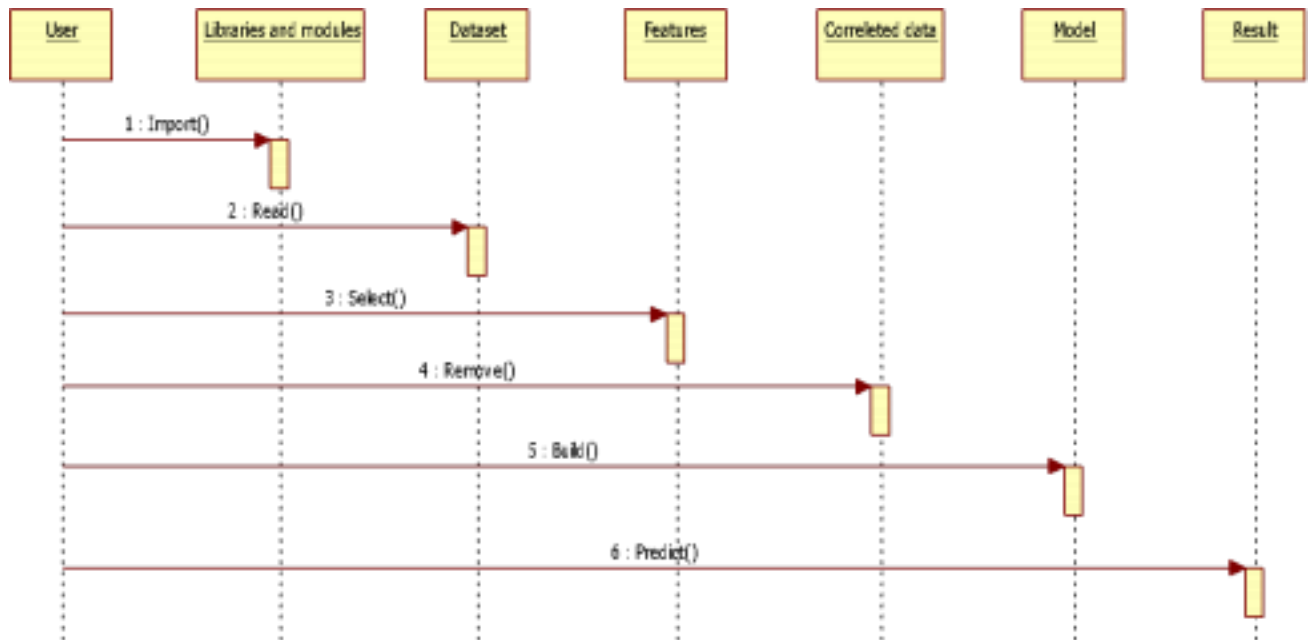
UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

UseCaseDiagram:

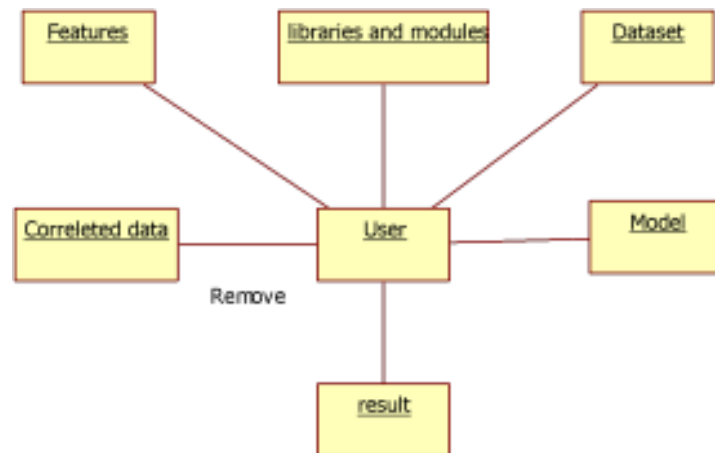


Sequence diagram:

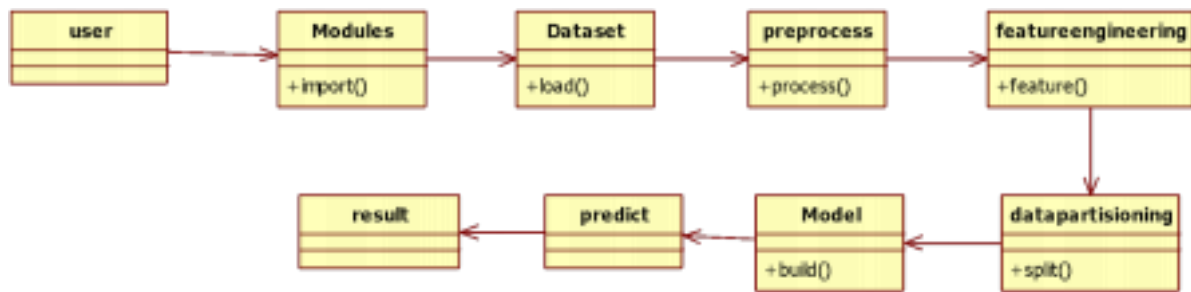
Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.



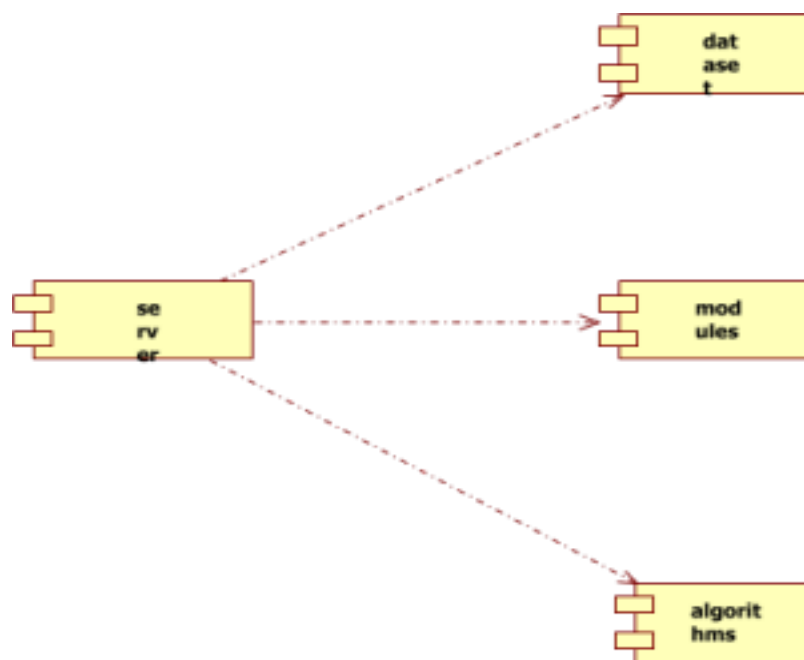
Collaboration Diagram:



Class Diagram:



Component Diagram



State Diagram:



TECHNOLOGY DESCRIPTION AND IMPLEMENTATION

6. TECHNOLOGY DESCRIPTION AND IMPLEMENTATION

Introduction to python framework

Introduction to Django This book is about Django, a web development framework that saves you time and makes web development a joy. Using Django. You can build and maintain high-quality web Application with minimal fuss. At its best, web development is an exciting act; at its worst, it can be a repetitive, frustration nuisance. Django lets you focus on the stuff – the crux of your web Application- while easing the pain of the repetitive bits. In doing so, it provides high-level abstractions of common Web development patterns, shorts cuts for frequent programming tasks, and clear conventions for how to solve problems. At the same time, Django tries to stay out of your way, letting you work. Outside the scope of the framework as needed. The goal of this book is to make you a Django expert. The focus is twofold, we explain, in depth, what Django does and how to build web applications with it. second, we discuss higher-level concepts where appropriate, answering the question “How can I Apply these effectively in my own project?” By reading this book, you’ll learn the skill needed to Develop powerful web site quickly, with code that is clean and easy to maintain.

What is a web Framework?

Django is a prominent member of a new generation of Web frameworks. So what exactly does that term mean? To answer that question, let’s consider the design of a Web application written using the Common Gateway Interface (CGI) standard, a popular way to write Web applications circa 1998. In those days, when you wrote a CGI application, you did everything yourself — the equivalent of baking a cake from scratch. For example, here’s a simple CGI script, written in Python, that displays the ten most recently published books from a database:

```

import MySQLdb

print "Content-Type: text/html"
print
print "<html><head><title>Books</title></head>"
print "<body>"
print "<h1>Books</h1>"
print "<ul>"

connection = MySQLdb.connect(user='me', passwd='letmein', db='my_db')
cursor = connection.cursor()
cursor.execute("SELECT name FROM books ORDER BY pub_date DESC LIMIT 10")
for row in cursor.fetchall():
    print "<li>%s</li>" % row[0]

print "</ul>"
print "</body></html>"

connection.close()

```

This code is straightforward. First, it prints a “Content-Type” line, followed by a blank line, as required by CGI. It prints some introductory HTML, connects to a database and executes a query that retrieves the latest ten books. Looping over those books, it generates an HTML unordered list. Finally, it prints the closing HTML and closes the database connection.

With a one-off dynamic page such as this one, the write-it-from-scratch approach isn’t necessarily bad. For one thing, this code is simple to comprehend — even a novice developer can read these 16 lines of Python and understand all it does, from start to finish. There’s nothing else to learn; no other code to read. It’s also simple to deploy: just save this code in a file called `latestbooks.cgi`, upload that file to a Web server, and visit that page with a browser. But as a Web application grows beyond the trivial, this approach breaks down, and you face a number of problems:

Should a developer really have to worry about printing the “Content-Type” line and remembering to close the database connection? This sort of boilerplate reduces programmer productivity and introduces opportunities for mistakes. These setup- and teardown-related tasks would best be handled by some common infrastructure.

✓ What happens when this code is reused in multiple environments, each with a separate database and password? At this point, some environment-specific configuration becomes essential.

- ✓ What happens when a Web designer who has no experience coding Python wishes to redesign

the page? Ideally, the logic of the page — the retrieval of books from the database — would be separate from the HTML display of the page, so that a designer could edit the latter without affecting the former.

- ✓ These problems are precisely what a Web framework intends to solve. A Web framework provides a programming infrastructure for your applications, so that you can focus on writing clean, maintainable code without having to reinvent the wheel. In a nutshell, that's what Django does.

Python

What Is A Script?

Up to this point, I have concentrated on the interactive programming capability of Python. This is a very useful capability that allows you to type in a program and to have it executed immediately in an interactive mode.

Scripts are reusable

Basically, a script is a text file containing the statements that comprise a Python program. Once you have created the script, you can execute it over and over without having to retype it each time.

Scripts are editable

Perhaps, more importantly, you can make different versions of the script by modifying the statements from one file to the next using a text editor. Then you can execute each of the individual versions. In this way, it is easy to create different programs with a minimum amount of typing.

You will need a text editor

Just about any text editor will suffice for creating Python script files.

You can use *Microsoft Notepad*, *Microsoft WordPad*, *Microsoft Word*, or just about any word processor if you want to.

Difference between a script and a program

Script:

Scripts are distinct from the core code of the application, which is usually written in a different language, and are often created or at least modified by the end-user. Scripts are often interpreted from source code or byte code, whereas the applications they control are traditionally compiled to native machine code.

Program:

The program has an executable form that the computer can use directly to execute the instructions.

The same program in its human-readable source code form, from which executable programs are derived(e.g., compiled).

Python

what is Python? Chances you are asking yourself this. You may have found this book because you want to learn to program but don't know anything about programming languages. Or you may have heard of programming languages like C, C++, C#, or Java and want to know what Python is and how it compares to "big name" languages. Hopefully I can explain it for you.

Python concepts

If your not interested in the the how and whys of Python, feel free to skip to the next chapter. In this chapter I will try to explain to the reader why I think Python is one of the best languages available and why it's a great one to start programming with.

- Open source general-purpose language.
- Object Oriented, Procedural, Functional
- Easy to interface with C/ObjC/Java/Fortran
- Easy-ish to interface with C++ (via SWIG)
- Great interactive environment

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.

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.

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 window 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.

Dynamic vs Static

Python is a dynamic-typed language. Many other languages are static typed, such as C/C++ and Java. A static typed language requires the programmer to explicitly tell the computer what type of “thing” each data value is.

For example, in C if you had a variable that was to contain the price of something, you would have to declare the variable as a “float” type.

This tells the compiler that the only data that can be used for that variable must be a floating point number, i.e. a number with a decimal point.

If any other data value was assigned to that variable, the compiler would give an error when trying

to compile the program.

Python, however, doesn't require this. You simply give your variables names and assign values to them. The interpreter takes care of keeping track of what kinds of objects your program is using. This also means that you can change the size of the values as you develop the program. Say you have another decimal number (a.k.a. a floating point number) you need in your program. With a static typed language, you have to decide the memory size the variable can take when you first initialize that variable. A double is a floating point value that can handle a much larger number than a normal float (the actual memory sizes depend on the operating environment).

If you declare a variable to be a float but later on assign a value that is too big to it, your program will fail; you will have to go back and change that variable to be a double. With Python, it doesn't matter. You simply give it whatever number you want and Python will take care of manipulating it as needed. It even works for derived values.

For example, say you are dividing two numbers. One is a floating point number and one is an integer. Python realizes that it's more accurate to keep track of decimals so it automatically calculates the result as a floating point number

Variables

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.

Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Standard Data Types

The data stored in memory can be of many types. For example, a person's age is stored as a numeric value and his or her address is stored as alphanumeric characters. Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.

Python has five standard data types –

- Numbers
- String
- List
- Tuple
- Dictionary

Python Numbers

Number data types store numeric values. Number objects are created when you assign a value to them

Python Strings

Strings in Python are identified as a contiguous set of characters represented in the quotation marks. Python allows for either pairs of single or double quotes. Subsets of strings can be taken using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.

Python Lists

Lists are the most versatile of Python's compound data types. A list contains items separated by commas and enclosed within square brackets ([]). To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data type. The values stored in a list can be accessed using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

Python Tuples

A tuple is another sequence data type that is similar to the list. A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses. The main differences between lists and tuples are: Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated. Tuples can be thought of as **read-only** lists.

Python Dictionary

Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are

usually numbers or strings. Values, on the other hand, can be any arbitrary Python object. Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).

Different modes in python

Python has two basic modes: normal and interactive. The normal mode is the mode where the scripted and finished .py files are run in the Python interpreter. Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole

Python libraries

1. Requests. The most famous http library written by kenneth reitz. It's a must have for every python developer.
2. Scrapy. If you are involved in webscraping then this is a must have library for you. After using this library you won't use any other.
3. wxPython. A GUI toolkit for python. I have primarily used it in place of tkinter. You will really love it.
4. Pillow. A friendly fork of PIL (Python Imaging Library). It is more user friendly than PIL and is a must have for anyone who works with images.
5. SQLAlchemy. A database library. Many love it and many hate it. The choice is yours.
6. BeautifulSoup. I know it's slow but this xml and html parsing library is very useful for beginners.
7. Twisted. The most important tool for any network application developer. It has a very beautiful api and is used by a lot of famous python developers.
8. NumPy. How can we leave this very important library? It provides some advance math functionalities to python.
9. SciPy. When we talk about NumPy then we have to talk about scipy. It is a library of algorithms and mathematical tools for python and has caused many scientists to switch from ruby to python.
10. Matplotlib. A numerical plotting library. It is very useful for any data scientist or any data analyzer.
11. Pygame. Which developer does not like to play games and develop them? This library will help you achieve your goal of 2d game development.

12. Pyglet. A 3d animation and game creation engine. This is the engine in which the famous python port of minecraft was made
13. PyQt. A GUI toolkit for python. It is my second choice after wxpython for developing GUI's for my python scripts.
14. PyGtk. Another python GUI library. It is the same library in which the famous Bittorrent client is created.
15. Scapy. A packet sniffer and analyzer for python made in python.
16. Pywin32. A python library which provides some useful methods and classes for interacting with windows.
17. Nltk. Natural Language Toolkit – I realize most people won't be using this one, but its generic enough. It is a very useful library if you want to manipulate strings. But its capacity is beyond that. Do check it out.
18. Nose. A testing framework for python. It is used by millions of python developers. It is a must have if you do test driven development.
19. SymPy. SymPy can do algebraic evaluation, differentiation, expansion, complex numbers, etc. It is contained in a pure Python distribution.
20. IPython. I just can't stress enough how useful this tool is. It is a python prompt on steroids. It has completion, history, shell capabilities, and a lot more. Make sure that you take a look at it.

Numpy

NumPy's main object is the homogeneous multidimensional array. It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In NumPy dimensions are called *axes*. The number of axes is *rank*.

- Offers Matlab-ish capabilities within Python
- Fast array operations
- 2D arrays, multi-D arrays, linear algebra etc.

Matplotlib

- High quality plotting library.

Python class and objects

These are the building blocks of OOP. Class creates a new object. This object can be anything, whether an abstract data concept or a model of a physical object, e.g. a chair. Each class has individual characteristics unique to that class, including variables and methods. Classes are very powerful and currently “the big thing” in most programming languages. Hence, there are several chapters dedicated to OOP later in the book. The class is the most basic component of object-oriented programming. Previously, you learned how to use functions to make your program do something. Now will move into the big, scary world of Object-Oriented Programming (OOP). To be honest, it took me several months to get a handle on objects. When I first learned C and C++, I did great; functions just made sense for me. Having messed around with BASIC in the early '90s, I realized functions were just like subroutines so there wasn't much new to learn. However, when my C++ course started talking about objects, classes, and all the new features of OOP, my grades definitely suffered. Once you learn OOP, you'll realize that it's actually a pretty powerful tool. Plus many Python libraries and APIs use classes, so you should at least be able to understand what the code is doing. One thing to note about Python and OOP: it's not mandatory to use objects in your code in a way that works best; maybe you don't need to have a full-blown class with initialization code and methods to just return a calculation. With Python, you can get as technical as you want. As you've already seen, Python can do just fine with functions. Unlike languages such as Java, you aren't tied down to a single way of doing things; you can mix functions and classes as necessary in the same program. This lets you build the codeObjects are an encapsulation of variables and functions into a single entity. Objects get their variables and functions from classes. Classes are essentially a template to create your objects.

Here's a brief list of Python OOP ideas:

- The class statement creates a class object and gives it a name. This creates a new namespace. • Assignments within the class create class attributes. These attributes are accessed by qualifying the name using dot syntax: `ClassName.Attribute`.
- Class attributes export the state of an object and its associated behavior. These attributes are shared by all instances of a class.
- Calling a class (just like a function) creates a new instance of the class. This is where the multiple copies part comes in.
- Each instance gets ("inherits") the default class attributes and gets its own namespace. This prevents instance objects from overlapping and confusing the program.
- Using the term `self` identifies a particular instance, allowing for per-instance attributes. This

allows items such as variables to be associated with a particular instance.

Inheritance

First off, classes allow you to modify a program without really making changes to it. To elaborate, by subclassing a class, you can change the behavior of the program by simply adding new components to it rather than rewriting the existing components. As we've seen, an instance of a class inherits the attributes of that class. However, classes can also inherit attributes from other classes. Hence, a subclass inherits from a superclass allowing you to make a generic superclass that is specialized via subclasses. The subclasses can override the logic in a superclass, allowing you to change the behavior of your classes without changing the superclass at all.

Operator Overloads

Operator overloading simply means that objects that you create from classes can respond to actions (operations) that are already defined within Python, such as addition, slicing, printing, etc. Even though these actions can be implemented via class methods, using overloading ties the behavior closer to Python's object model and the object interfaces are more consistent to Python's built-in objects, hence overloading is easier to learn and use. User-made classes can override nearly all of Python's built-in operation methods

Exceptions

I've talked about exceptions before but now I will talk about them in depth. Essentially, exceptions are events that modify program's flow, either intentionally or due to errors. They are special events that can occur due to an error, e.g. trying to open a file that doesn't exist, or when the program reaches a marker, such as the completion of a loop. Exceptions, by definition, don't occur very often; hence, they are the "exception to the rule" and a special class has been created for them. Exceptions are everywhere in Python. Virtually every module in the standard Python library uses them, and Python itself will raise them in a lot of different circumstances.

Here are just a few examples:

- Accessing a non-existent dictionary key will raise a **Key Error** exception.
- Searching a **list** for a non-existent value will raise a **Value Error** exception
- Calling a non-existent method will raise an **Attribute**

Error exception.

- Referencing a non-existent variable will raise a Name Error exception.
- Mixing datatypes without coercion will raise a Type Error exception.

One use of exceptions is to catch a fault and allow the program to continue working; we have seen this before when we talked about files. This is the most common way to use exceptions. When programming with the Python command line interpreter, you don't need to worry about catching exceptions. Your program is usually short enough to not be hurt too much if an exception occurs. Plus, having the exception occur at the command line is a quick and easy way to tell if your code logic has a problem. However, if the same error occurred in your real program, it will fail and stop working. Exceptions can be created manually in the code by raising an exception. It operates exactly as a system-caused exceptions, except that the programmer is doing it on purpose. This can be for a number of reasons. One of the benefits of using exceptions is that, by their nature, they don't put any overhead on the code processing. Because exceptions aren't supposed to happen very often, they aren't processed until they occur. Exceptions can be thought of as a special form of the if/elif statements. You can realistically do the same thing with if blocks as you can with exceptions. However, as already mentioned, exceptions aren't processed until they occur; if blocks are processed all the time. Proper use of exceptions can help the performance of your program. The more infrequent the error might occur, the better off you are to use exceptions; using if blocks requires Python to always test extra conditions before continuing. Exceptions also make code management easier: if your programming logic is mixed in with error-handling if statements, it can be difficult to read, modify, and debug your program.

User-Defined Exceptions I won't spend too much time talking about this, but Python does allow for a programmer to create his own exceptions. You probably won't have to do this very often but it's nice to have the option when necessary. However, before making your own exceptions, make sure there isn't one of the built-in exceptions that will work for you. They have been "tested by fire" over the years and not only work effectively, they have been optimized for performance and are bug-free. Making your own exceptions involves object-oriented programming, which will be covered in the next chapter. To make a custom exception, the programmer determines which base exception to use as the class to inherit from, e.g. making an exception for negative numbers or one for imaginary numbers would probably fall under the Arithmetic Error exception class. To make a custom exception, simply inherit the base exception and define what it will do.

Python modules

Python allows us to store our code in files (also called modules). This is very useful for more serious programming, where we do not want to retype a long function definition from the very beginning just to change one mistake. In doing this, we are essentially defining our own modules, just like the modules defined already in the Python library.

To support this, Python has a way to put definitions in a file and use them in a script or in an interactive instance of the interpreter. Such a file is called a *module*; definitions from a module can be *imported* into other modules or into the *main* module.

Testing code

As indicated above, code is usually developed in a file using an editor. To test the code, import it into a Python session and try to run it. Usually there is an error, so you go back to the file, make a correction, and test again. This process is repeated until you are satisfied that the code works. The entire process is known as the development cycle. There are two types of errors that you will encounter. Syntax errors occur when the form of some command is invalid. This happens when you make typing errors such as misspellings, or call something by the wrong name, and for many other reasons. Python will always give an error message for a syntax error.

Functions in Python

It is possible, and very useful, to define our own functions in Python. Generally speaking, if you need to do a calculation only once, then use the interpreter. But when you or others have need to perform a certain type of calculation many times, then define a function.

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task. To carry out that specific task, the function might or might not need multiple inputs. When the task is carried out, the function can or cannot return one or more values.

There are three types of functions in python:

help(),min(),print().

Python Namespace

Generally speaking, a **namespace** (sometimes also called a context) is a naming system for making names unique to avoid ambiguity. Everybody knows a namespacing system from daily life, i.e. the naming of people in firstname and family name (surname).

An example is a network: each network device (workstation, server, printer, ...) needs a unique

name and address. Yet another example is the directory structure of file systems. The same file name can be used in different directories, the files can be uniquely accessed via the pathnames.

Many programming languages use namespaces or contexts for identifiers. An identifier defined in a namespace is associated with that namespace.

This way, the same identifier can be independently defined in multiple namespaces. (Like the same file names in different directories) Programming languages, which support namespaces, may have different rules that determine to which namespace an identifier belongs. Namespaces in Python are implemented as Python dictionaries, this means it is a mapping from names (keys) to objects (values). The user doesn't have to know this to write a Python program and when using namespaces.

Some namespaces in Python:

- **global names** of a module
- **local names** in a function or method invocation
- **built-in names**: this namespace contains built-in functions (e.g. `abs()`, `cmp()`, ...) and built-in exception names

Garbage Collection

Garbage Collector exposes the underlying memory management mechanism of Python, the automatic garbage collector. The module includes functions for controlling how the collector operates and to examine the objects known to the system, either pending collection or stuck in reference cycles and unable to be freed.

Python XML Parser

XML is a portable, open source language that allows programmers to develop applications that can be read by other applications, regardless of operating system and/or developmental language. What is XML? The Extensible Markup Language XML is a markup language much like HTML or SGML.

This is recommended by the World Wide Web Consortium and available as an open standard. XML is extremely useful for keeping track of small to medium amounts of data without requiring a SQL-based backbone.

XML Parser Architectures and APIs The Python standard library provides a minimal but useful set of interfaces to work with XML.

The two most basic and broadly used APIs to XML data are the SAX and DOM interfaces. Simple API for XML SAX : Here, you register callbacks for events of interest and then let the parser proceed through the document.

This is useful when your documents are large or you have memory limitations, it parses the file as it reads it from disk and the entire file is never stored in memory.

Document Object Model DOM API : This is a World Wide Web Consortium recommendation wherein the entire file is read into memory and stored in a hierarchical tree – based form to represent all the features of an XML document.

SAX obviously cannot process information as fast as DOM can when working with large files. On the other hand, using DOM exclusively can really kill your resources, especially if used on a lot of small files.

SAX is read-only, while DOM allows changes to the XML file. Since these two different APIs literally complement each other, there is no reason why you cannot use them both for large projects.

Python Web Frameworks

A web framework is a code library that makes a developer's life easier when building reliable, scalable and maintainable web applications.

Why are web frameworks useful?

Web frameworks encapsulate what developers have learned over the past twenty years while programming sites and applications for the web. Frameworks make it easier to reuse code for common HTTP operations and to structure projects so other developers with knowledge of the framework can quickly build and maintain the application.

Common web framework functionality

Frameworks provide functionality in their code or through extensions to perform common operations required to run web applications. These common operations include:

1. URL routing
2. HTML, XML, JSON, and other output format templating
3. Database manipulation
4. Security against Cross-site request forgery (CSRF) and other attacks
5. Session storage and retrieval

Not all web frameworks include code for all of the above functionality. Frameworks fall on the spectrum from executing a single use case to providing every known web framework feature to every developer. Some frameworks take the "batteries-included" approach where everything possible comes bundled with the framework while others have a minimal core package that is amenable to extensions provided by other packages.

Comparing web frameworks

There is also a repository called [compare-python-web-frameworks](#) where the same web application is being coded with varying Python web frameworks, templating engines and object. Web framework resources

- When you are learning how to use one or more web frameworks it's helpful to have an idea of what the code under the covers is doing.
- Frameworks is a really well done short video that explains how to choose between web frameworks. The author has some particular opinions about what should be in a framework. For the most part I agree although I've found sessions and database ORMs to be a helpful part of a framework when done well.
- what is a web framework? is an in-depth explanation of what web frameworks are and their relation to web servers.
- Django vs Flask vs Pyramid: Choosing a Python web framework contains background information and code comparisons for similar web applications built in these three big Python frameworks.
- This fascinating blog post takes a look at the code complexity of several Python web frameworks by providing visualizations based on their code bases.
- Python's web frameworks benchmarks is a test of the responsiveness of a framework with encoding an object to JSON and returning it as a response as well as retrieving data from the database and rendering it in a template. There were no conclusive results but the output is fun to read about nonetheless.
- What web frameworks do you use and why are they awesome? is a language agnostic Reddit discussion on web frameworks. It's interesting to see what programmers in other languages like and dislike about their suite of web frameworks compared to the main Python frameworks.
- This user-voted question & answer site asked "What are the best general purpose Python web frameworks usable in production?". The votes aren't as important as the list of the many frameworks that are available to Python developers.

Web frameworks learning checklist

1. Choose a major Python web framework (Django or Flask are recommended) and stick with it. When you're just starting it's best to learn one framework first instead of bouncing around trying to understand every framework.
2. Work through a detailed tutorial found within the resources links on the framework's page(ii)

CODING

7.CODING

```
#Import the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import diffprivlib.models as dp
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
```

#Reading the datasets

```
data = pd.read_csv('breast_cancer.csv')
data.head()
```

ii

#Selecting the best features

```
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
y = df['diagnosis']
X = df.drop(['diagnosis','Unnamed: 32'],axis = 1)
#apply SelectKBest class to extract top 10 best features
bestfeatures = SelectKBest(score_func=chi2, k=20)
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
#concat two dataframes for better visualization
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Specs','Score'] #naming the dataframe columns
print(featureScores.nlargest(20,'Score')) #print 20 best features
#splitting the data into training and testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.3, random_state = 42)
```

#Training the model

```
from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
lr = Pipeline([
    ('scaler', MinMaxScaler()),
    ('clf', dp.LogisticRegression(epsilon=10))
])
```

```
lr.fit(X_train, y_train)
```

#Predicting the output and checking the accuracy

```
from sklearn.metrics import accuracy_score
y_pred = lr.predict(X_test)
print('Acc :',accuracy_score(y_test,y_pred))
```

#Finding the correleated features

```
col_corr = set()
corr_mat = df.corr()
for i in range(len(corr_mat.columns)):
    for j in range(i):
        if ((corr_mat.iloc[i,j]) > 0.9):
            colname = corr_mat.columns[i]
            col_corr.add(colname)
```

#Removing correleated features

```
df.drop(['area_mean','area_se','area_worst','concave points_mean',
'concave points_worst','perimeter_mean','perimeter_se',
'perimeter_worst','radius_worst','texture_worst'], axis = 1, inplace = True)
```

#Applying the differential privacy algorithm

```
dp_lr = Pipeline([
    ('scaler', MinMaxScaler()),
    ('clf', dp.LogisticRegression(epsilon=10))
])
dp_lr.fit(X_train, y_train)
```

#Finding output and accuracy

```
from sklearn.metrics import accuracy_score
y_pred = dp_lr.predict(X_test)
print('Acc :',accuracy_score(y_test,y_pred))
```

TESTING AND TEST CASES

8.1 INTRODUCTION TO TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant costs associated with a software failure are motivating factors for we planned, through testing. Testing is the process of executing a program with the intent of finding an error. The design of tests for software and other engineered products can be as challenging as the initial design of the product itself.

There of basically **two types of testing** approaches.

One is ***Black-Box testing*** – the specified function that a product has been designed to perform, tests can be conducted that demonstrate each function is fully operated. The other is ***White-Box testing*** – knowing the internal workings of the product ,tests can be conducted to ensure that the

internal operation of the product performs according to specifications and all internal components have been adequately exercised. White box and Black box testing methods have been used to test this package. The entire loop constructs have been tested for their boundary and intermediate conditions. The test data was designed with a view to check for all the conditions and logical decisions. Error handling has been taken care of by the use of exception handlers.

8.2 TESTING STRATEGIES:

Testing is a set of activities that can be planned in advanced and conducted systematically. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high-level tests that validate major system functions against customer requirements.

Software testing is one element of verification and validation. Verification refers to the set of activities that ensure that software correctly implements a specific function. Validation refers to a different set of activities that ensure that the software that has been built is traceable to customer requirements.

The main objective of software testing is to uncover errors. To fulfill this objective, a series of test steps unit, integration, validation and system tests are planned and executed. Each test step is accomplished through a series of systematic test techniques that assist in the design of test cases. With each testing step, the level of abstraction with which software is considered is broadened.

Testing is the only way to assure the quality of software and it is an umbrella activity rather than a separate phase. This is an activity to be performed in parallel with the software effort and one that consists of its own phases of analysis, design, implementation, execution and maintenance.

UNIT TESTING:

This testing method considers a module as a single unit and checks the unit at interfaces and communicates with other modules rather than getting into details at statement level. Here the module will be treated as a black box, which will take some input and generate output. Outputs for a given set of input combinations are pre-calculated and are generated by the module.

SYSTEM TESTING:

Here all the pre-tested individual modules will be assembled to create the larger system and tests are carried out at system level to make sure that all modules are working in synchronism with each other. This testing methodology helps in making sure that all modules which are running perfectly

when checked individually are also running in cohesion with other modules. For this testing we create test cases to check all modules once and then generated test combinations of test paths throughout the system to make sure that no path is making its way into chaos.

INTEGRATED TESTING

Testing is a major quality control measure employed during software development. Its basic function is to detect errors. Sub functions when combined may not produce than it is desired. Global data structures can represent the problems. Integrated testing is a systematic technique for constructing the program structure while conducting the tests. To uncover errors that are associated with interfacing the objective is to make unit test modules and built a program structure that has been detected by design. In a non - incremental integration all the modules are combined in advance and the program is tested as a whole. Here errors will appear in an end less

loop function. In incremental testing the program is constructed and tested in small segments where the errors are isolated and corrected.

Different incremental integration strategies are top – down integration, bottom – up integration, regression testing.

TOP-DOWN INTEGRATION TEST

Modules are integrated by moving downwards through the control hierarchy beginning with main program. The subordinate modules are incorporated into structure in either a breadth first manner or depth first manner. This process is done in five steps:

- Main control module is used as a test driver and steps are substituted or all modules directly to main program.
- Depending on the integration approach selected subordinate is replaced at a time with actual modules.
- Tests are conducted.
- On completion of each set of tests another stub is replaced with the real module
- Regression testing may be conducted to ensure trha5t new errors have not been introduced. This process continuous from step 2 until entire program structure is reached. In top down integration strategy decision making occurs at upper levels in the hierarchy and is encountered first. If major control problems do exists early recognitions is essential.

If depth first integration is selected a complete function of the software may be implemented and

demonstrated.

Some problems occur when processing at low levels in hierarchy is required to adequately test upper level steps to replace low-level modules at the beginning of the top down testing. So no data flows upward in the program structure.

BOTTOM-UP INTEGRATION TEST

Begins construction and testing with atomic modules. As modules are integrated from the bottom up, processing requirement for modules subordinate to a given level is always available and need for stubs is eliminated. The following steps implements this strategy.

- Low-level modules are combined in to clusters that perform a specific software sub function.
- A driver is written to coordinate test case input and output.
- Cluster is tested.
- Drivers are removed and moving upward in program structure combines clusters.

Integration moves upward, the need for separate test driver's lesions.

If the top levels of program structures are integrated top down, the number of drivers can be reduced substantially and integration of clusters is greatly simplified.

REGRESSION TESTING

Each time a new module is added as a part of integration as the software changes. Regression testing is an actually that helps to ensure changes that do not introduce unintended behavior as additional errors.

Regression testing maybe conducted manually by executing a subset of all test cases or using automated capture play back tools enables the software engineer to capture the test case and results for subsequent playback and compression. The regression suit contains different classes of test cases.

A representative sample to tests that will exercise all software functions.

Additional tests that focus on software functions that are likely to be affected by the change.

8.3 IMPLEMENTATION

Implementation is the process of converting a new or revised system design into operational one.

There are three types of Implementation:

Implementation of a computer system to replace a manual system. The problems encountered

are converting files, training users, and verifying printouts for integrity.

Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned there can be many problems.

Implementation of a modified application to replace an existing one using the same computer.

This type of conversion is relatively easy to handle, provided there are no major changes in the files.

Implementation in Generic tool project is done in all modules. In the first module User level identification is done. In this module every user is identified whether they are genuine one or not to access the database and also generates the session for the user. Illegal use of any form is strictly avoided.

In the Table creation module, the tables are created with user specified fields and user can create many table at a time. They may specify conditions, constraints and calculations in creation of tables. The Generic code maintain the user requirements through out the project. In Updating module user can update or delete or Insert the new record into the database. This is very important module in Generic code project. User has to specify the filed value in the form then the Generic tool automatically gives whole filed values for that particular record. In Reporting module user can get the reports from the database in 2Dimentional or 3Dimensional view. User has to select the table and specify the condition then the report will be generated for the user.

INPUT AND OUTPUT DESIGN

9. INPUT AND OUTPUT DESIGN

9.1 INPUT AND OUTPUT

The following some are the projects inputs and outputs.

Inputs:

Importing the all required packages like numpy, pandas, matplotlib, scikit – learn and required machine learning algorithms packages.

Setting the dimensions of visualization graph.

Downloading and importing the dataset and convert to data frame.

Outputs:

Preprocessing the importing data frame for imputing nulls with the related information.

All are displaying cleaned outputs.

After applying machine learning algorithms it will give good results and visualization plots.

INPUT DESIGN

Input design is a part of overall system design. The main objective during the input design is

as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization,
- Internal Outputs whose destination is within organization and they are the ▪

User's main interface with the computer.

- Operational outputs whose use is purely within the computer department.
- Interface outputs, which involve the user in communicating directly with The outputs were needed to be generated as a hard copy and as well as queries to be viewed on the screen. Keeping in view these outputs, the format for the output is taken from the outputs, which are currently being obtained after manual processing. The standard printer is to be used as output media for hard copies.

OUT PUT SCREEN

10. OUTPUT SCREENS

```

from sklearn.linear_model import LogisticRegression
import warnings
warnings.filterwarnings('ignore')
lr = Pipeline([
    ('scaler', MinMaxScaler()),
    ('clf', dp.LogisticRegression(epsilon=5))
])

```

```
lr.fit(X_train, y_train)
```

```

Pipeline(memory=None,
       steps=[('scaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
              ('clf',
               LogisticRegression(C=1.0, data_norm=3.0003767724634693,
                                  epsilon=5, fit_intercept=True, max_iter=100,
                                  n_jobs=None, tol=0.0001, verbose=0,
                                  warm_start=False))],
       verbose=False)

```

```

from sklearn.metrics import accuracy_score
y_pred_lr = lr.predict(X_test)
print('Acc : ',accuracy_score(y_test,y_pred_lr))

```

```
Acc : 0.8596491228070176
```

```

dp_lr = Pipeline([
    ('scaler', MinMaxScaler()),
    ('clf', dp.LogisticRegression(epsilon=4))
])

```

```
dp_lr.fit(X_train, y_train)
```

```

Pipeline(memory=None,
       steps=[('scaler', MinMaxScaler(copy=True, feature_range=(0, 1))),
              ('clf',
               LogisticRegression(C=1.0, data_norm=2.821762643727689,
                                  epsilon=4, fit_intercept=True, max_iter=100,
                                  n_jobs=None, tol=0.0001, verbose=0,
                                  warm_start=False))],
       verbose=False)

```

```

from sklearn.metrics import accuracy_score
y_pred_dp_lr = dp_lr.predict(X_test)
print('Acc : ',accuracy_score(y_test,y_pred_dp_lr))

```

```
Acc : 0.8596491228070176
```

FUTURE ENHANCEMENTS

11. FUTURE ENHANCEMENTS

The selected features can have less correlation across the whole dataset and achieve a desirable accuracy results. For data queries, the correlation in the whole dataset is also reduced with the proposed scheme. However, as we noted the MAE is not always better than other schemes. This is because the sensitivity is related the type of queries and dataset itself. Deleted or added features in the dataset can reduce the data correlation, which may bring in new error with regard to different queries.

12. CONCLUSION

In this paper, we identified the privacy issue of the data correlation in machine learning, which may result in more privacy loss than expected in industrial applications. We propose a novel feature selection scheme to reduce data correlation with little compromise to data utility. The proposed CR-FS(correlated feature selection) scheme includes steps that consider the accuracy of predicted results, the privacy preserving and the data correlation in the dataset. Our proposed algorithm strikes a better trade-off between data utility and privacy leaks for correlated datasets. The method's performance is evaluated via extensive experiments, and the results prove that our proposed CR-FS scheme provides better data utility for both data analysis and data queries compared to traditional schemes.

BIBLIOGRAPHY

13. BIBLIOGRAPHY

For software installation:

<https://www.anaconda.com/download/>

<https://www.python.org/downloads/release/python-360/>

Modules:

- ✓ Install numpy
- ✓ Install pandas
- ✓ Install matplotlib
- ✓ Install scikit – learn

References:

- [1] U.S.Shanthamallu,A.Spanias,C.TepedelenliogluandM.Stanley,“Abrief survey of machine learning methods and their sensor and IoT applications,” In 2017 8th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1-8.
- [2] I.A.T. Hashem, V. Chang, N.B. Anuar, K. Adewole, I. aqoob, A. Gani, E. Ahmed and H. Chiroma, “The role of big data in smart city,” International Journal of Information Management, 36(5), pp.748-758.
- [3] C. Yin, J. Xi, R. Sun and J. Wang, “Location privacy protection based on differential privacy strategy for big data in industrial internet of things,” IEEE Transactions on Industrial Informatics, 2017, 14(8), pp.3628-3636.
- [4] A. Solanas, C. Patsakis, M. Conti, I. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. Perez-Martinez, R. Pietro, D. Perrea, “Smart health: a context-aware health paradigm within smart cities,” IEEE Communications Magazine, vol. 52, no. 8, pp. 74–81. [5] C.M. Benjamin, M. Fung, K. Wang, R. Chen and P.S. Yu, “Privacy- preserving data publishing: A survey of recent developments,” ACM Computing Surveys, 2010, 42(4), pp.1-53. [6] C. Dwork, 2006, “Differential privacy,” in ICALP, pp. 1–12.
- [7] M. Yang, T. Zhu, Y. Xiang and W. Zhou, 2018. “Density-based location preservation for mobile crowdsensing with differential privacy,” IEEE Access, 2018, 6, pp.14779-14789. [8] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, “PPFA: privacy preserving fog-enabled aggregation in smart grid,” IEEE Transactions on Industrial Informatics, 2018, 14(8), pp.3733-3744.
- [9] Y. Liu, W. Guo, C.I. Fan, L. Chang and C. Cheng, “A practical privacy- preserving data aggregation (3PDA) scheme for smart grid,” IEEE Transactions on Industrial Informatics, 2019, 15(3), pp.1767-1774.
- [10] D. Ye, T. Zhu, W. Zhou, and P.S. Yu, ”Differentially Private Malicious Agent Avoidance in Multiagent Advising Learning,” IEEE transactions on cybernetics, 2019, DOI:10.1109/TCYB.2019.2906574.

Reference Papers

16. Research papers

- [1] U.S. Shanthamallu, A. Spanias, C. Tepedelenlioglu and M. Stanley, "A brief survey of machine learning methods and their sensor and IoT applications," In 2017 8th International Conference on Information, Intelligence, Systems and Applications (IISA), pp. 1-8.
- [2] I.A.T. Hashem, V. Chang, N.B. Anuar, K. Adewole, I. Aqoob, A. Gani, E. Ahmed and H. Chiroma, "The role of big data in smart city," International Journal of Information Management, 36(5), pp. 748-758.
- [3] C. Yin, J. Xi, R. Sun and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial internet of things," IEEE Transactions on Industrial Informatics, 2017, 14(8), pp. 3628-3636.
- [4] A. Solanas, C. Patsakis, M. Conti, I. Vlachos, V. Ramos, F. Falcone, O. Postolache, P. Perez-Martinez, R. Pietro, D. Perrea, "Smart health: a context-aware health paradigm within smart cities," IEEE Communications Magazine, vol. 52, no. 8, pp. 74-81.
- [5] C.M. Benjamin, M. Fung, K. Wang, R. Chen and P.S. Yu, "Privacy-preserving data publishing: A survey of recent developments," ACM Computing Surveys, 2010, 42(4), pp. 1-53.
- [6] C. Dwork, 2006, "Differential privacy," in ICALP, pp. 1-12.
- [7] M. Yang, T. Zhu, Y. Xiang and W. Zhou, 2018. "Density-based location preservation for mobile crowdsensing with differential privacy," IEEE Access, 2018, 6, pp. 14779-14789.
- [8] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: privacy preserving fog-enabled aggregation in smart grid," IEEE Transactions on Industrial Informatics, 2018, 14(8), pp. 3733-3744.
- [9] Y. Liu, W. Guo, C.I. Fan, L. Chang and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," IEEE Transactions on Industrial Informatics, 2019, 15(3), pp. 1767-1774.
- [10] D. Ye, T. Zhu, W. Zhou, and P.S. Yu, "Differentially Private Malicious Agent Avoidance in Multiagent Advising Learning," IEEE transactions on cybernetics, 2019, DOI:10.1109/TCYB.2019.2906574.