SIX MONTH SOFTWARE TRAINING

ON

FACIAL SENTIMENT RECOGNIZER Using MACHINE LEARNING

at

Coding Ninjas

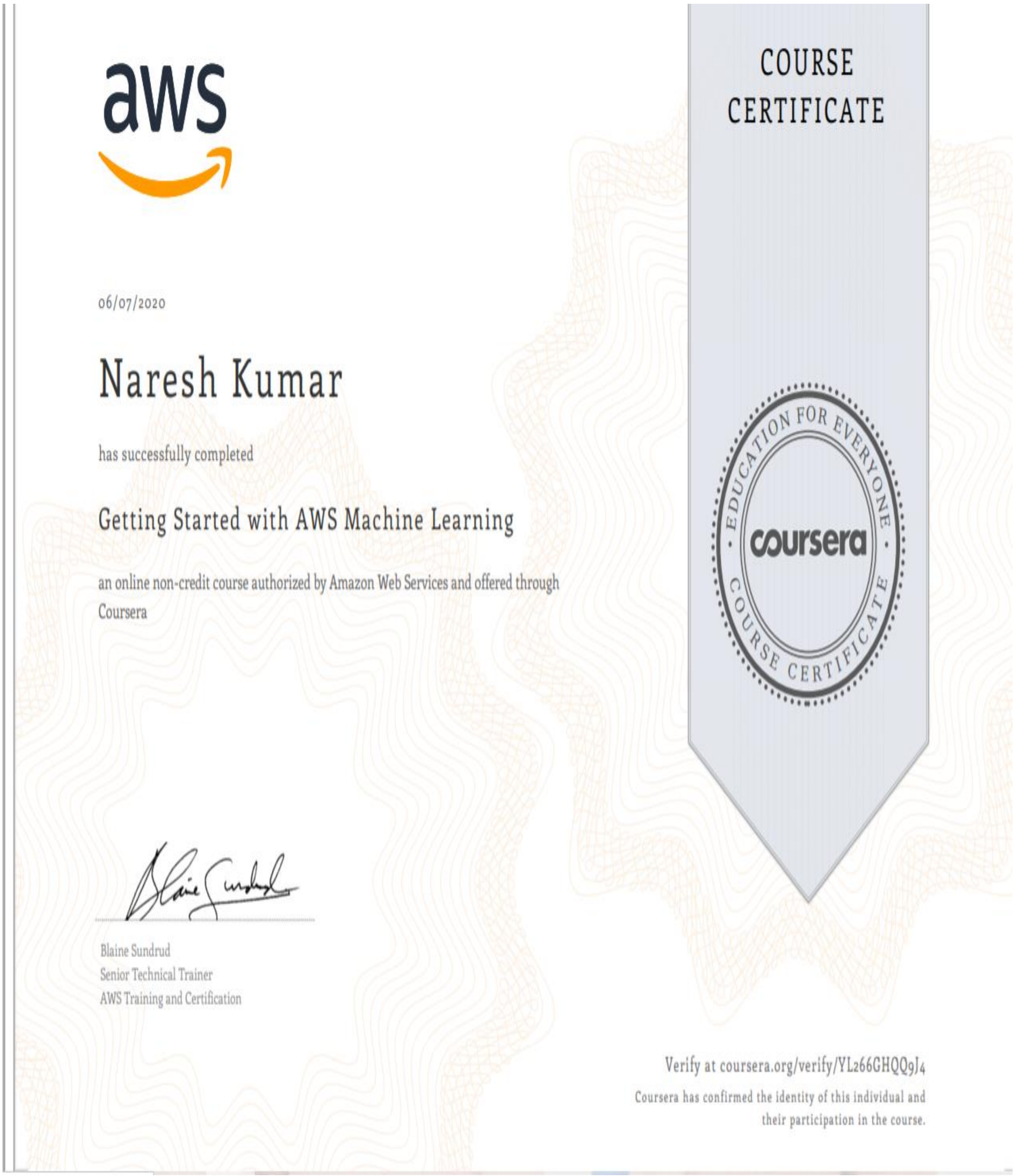
From jan-2020 to june-2020

SUBMITTED BY – **Laksh Kiran, Naresh Kumar**

ROLL NO. 1608960, 1608965

Department of Computer Science & Engineering

KHALSA COLLEGE OF ENGINEERING AND TECHNOLOGY
AMRITSAR



aws

COURSE CERTIFICATE

06/07/2020

Naresh Kumar

has successfully completed

Getting Started with AWS Machine Learning

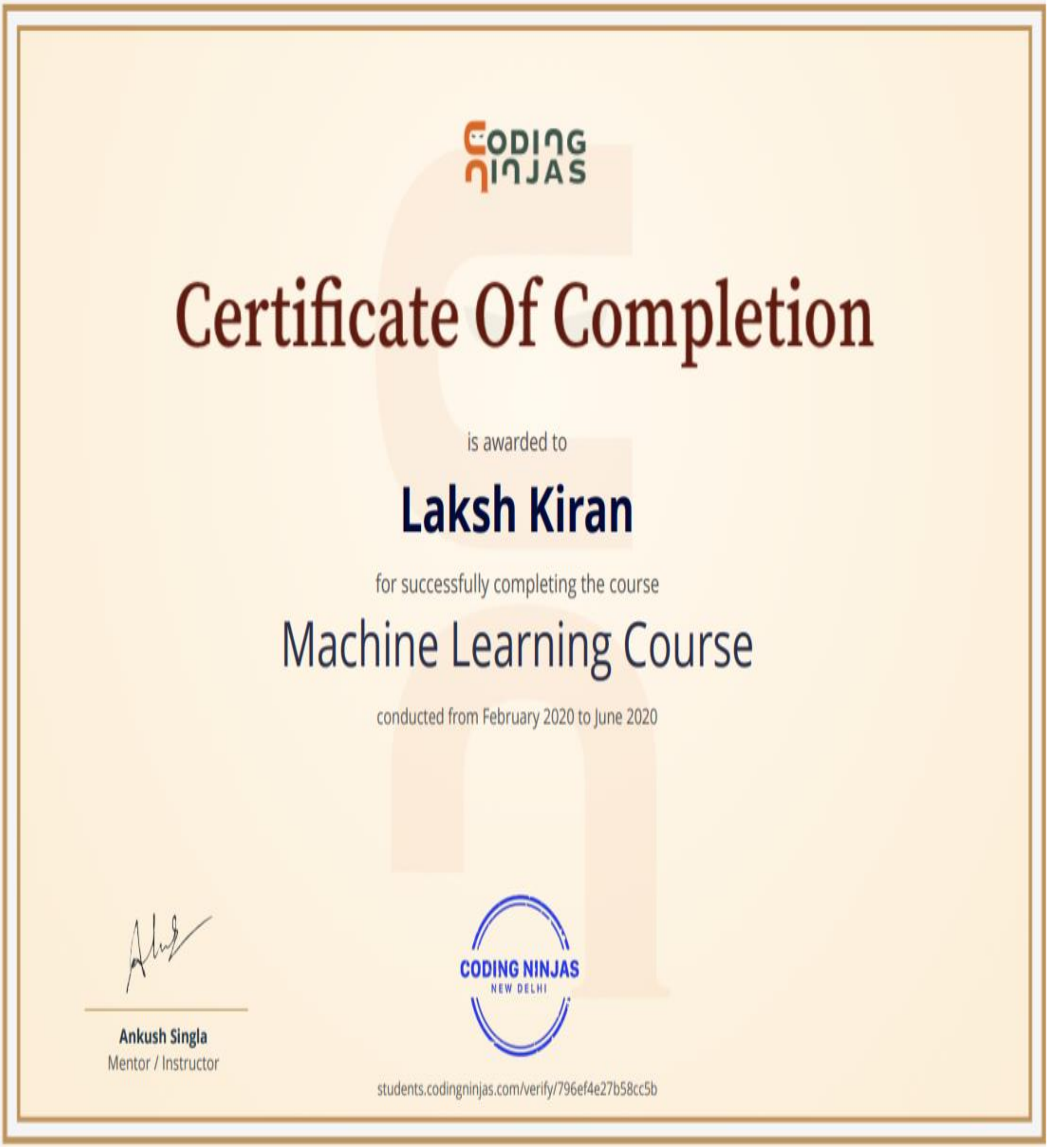
an online non-credit course authorized by Amazon Web Services and offered through Coursera

EDUCATION FOR EVERYONE
coursera
COURSE CERTIFICATE

Blaine Sundrud
Senior Technical Trainer
AWS Training and Certification

Verify at coursera.org/verify/YL266GHQQ9J4

Coursera has confirmed the identity of this individual and their participation in the course.



CODING NINJAS

Certificate Of Completion

is awarded to

Laksh Kiran

for successfully completing the course

Machine Learning Course

conducted from February 2020 to June 2020

Ankush Singla
Mentor / Instructor

CODING NINJAS
NEW DELHI

students.codingninjas.com/verify/796ef4e27b58cc5b

ACKNOWLEDGEMENT

This is a humble effort to express my sincere gratitude towards those who have guided and helped me to complete this project.

A project is major milestone during the study period of a student. As such this project was a challenge to me and was an opportunity to prove my calibrate. I am highly grateful and obliged to each and every one making me help out of problems being faced by me.

Last but not the least I am very thankful to our Head of Department **Dr.Kirandeep Singh** and **Er. Mannet Kaur** who gave us an opportunity to face real time problems while fulfilling need of an organization by making project for them.

Naresh Kumar

Laksh Kiran

ABSTRACT

There is rise in the amount of data on the internet and most of the data is unlabeled. There is plenty of the images which have no label or value attached with them. So, most of the data still not usable for any person. Images have great value associated with them if they correctly identified. For example, an image of person captured from the Store can tell about their experience in the store. So, we can identify the problem if exist with any of product and improve that. It is very costly to hire the people for this purpose. So, it will take valuable money of employer. So, our platform will do this work automatically. We can identify 7 Different Emotion of person by just having his/her face photo. This will allow employer to get data about the customer experience at large scale. There are plenty of use cases of the **Sentiment Emotion Recognizer.** So Now any one can know the sentiment of the human from our platform.

TABLE of CONTENTS

INTRODUCTION

1.1 Problem Definition

In present time, there is lot of data on internet which is difficult to identify. Also businesses find difficult to know the feedback of the customer for improvement of their business. More and More data every day being generated but it's difficult know the right sentiment of data or images. To know the feedback, lot of time and money had wasted.

1.2 Need of Study

As per problem, there is lot of time and resource can saved, If we can automate the emotion recognition. So, lot of study can be done in this field. This will lead to customer satisfaction in any business and real time feedback from customer so that business can improve their service.

1.3 Scope of Study

Facial Sentiment analysis is a image analysis method that detects emotion from any of image of person. There are basically 7 emotions- Angry, Disgust, Fear, Happy, Sad, Surprise, Neutral

Understanding people's emotions is essential for businesses since customers are able to express their thoughts and feelings more openly than ever before. By automatically analyzing customer feedback, from survey responses to social media conversations, brands are able to listen attentively to their customers, and tailor products and services to meet their needs. This type of sentiment analysis aims at detecting emotions, like happiness, frustration, anger, sadness, and so on.

Why Perform Sentiment Analysis?

It's estimated that 80% of the world's data is unstructured, in other words it's unorganized. Huge volumes of image data is created every day but it's hard to analyze, understand, and sort through, not to mention time-consuming and expensive.

Sentiment analysis, however, helps businesses make sense of all this unstructured text by automatically tagging it.

1.4 Benefits of Facial Sentiment Recognizer:

1. Sorting Data at Scale Can you imagine manually sorting through thousands of tweets, customer support conversations, or surveys? There’s just too much data to process manually. Sentiment analysis helps businesses process huge amounts of data in an efficient and cost-effective way.

2. Real-Time Analysis Sentiment analysis can identify critical issues in real-time, for example is a PR crisis on social media escalating? Is an angry customer about to churn? Sentiment analysis models can help you immediately identify these kinds of situations and gauge brand sentiment, so you can take action right away.

3. Consistent criteria It’s estimated that people only agree around 60-65% of the time when determining the sentiment of a particular image. sentiment is highly subjective, influenced by personal experiences, thoughts, and beliefs. By using a centralized sentiment analysis system, companies can apply the same criteria to all their data, helping them improve accuracy and gain better insights.

1.5 Sentiment Analysis Use Cases & Applications

Sentiment analysis applications:

- Social media monitoring
- Brand monitoring
- Voice of customer (VoC)
- Customer service
- Market research

ER Diagram

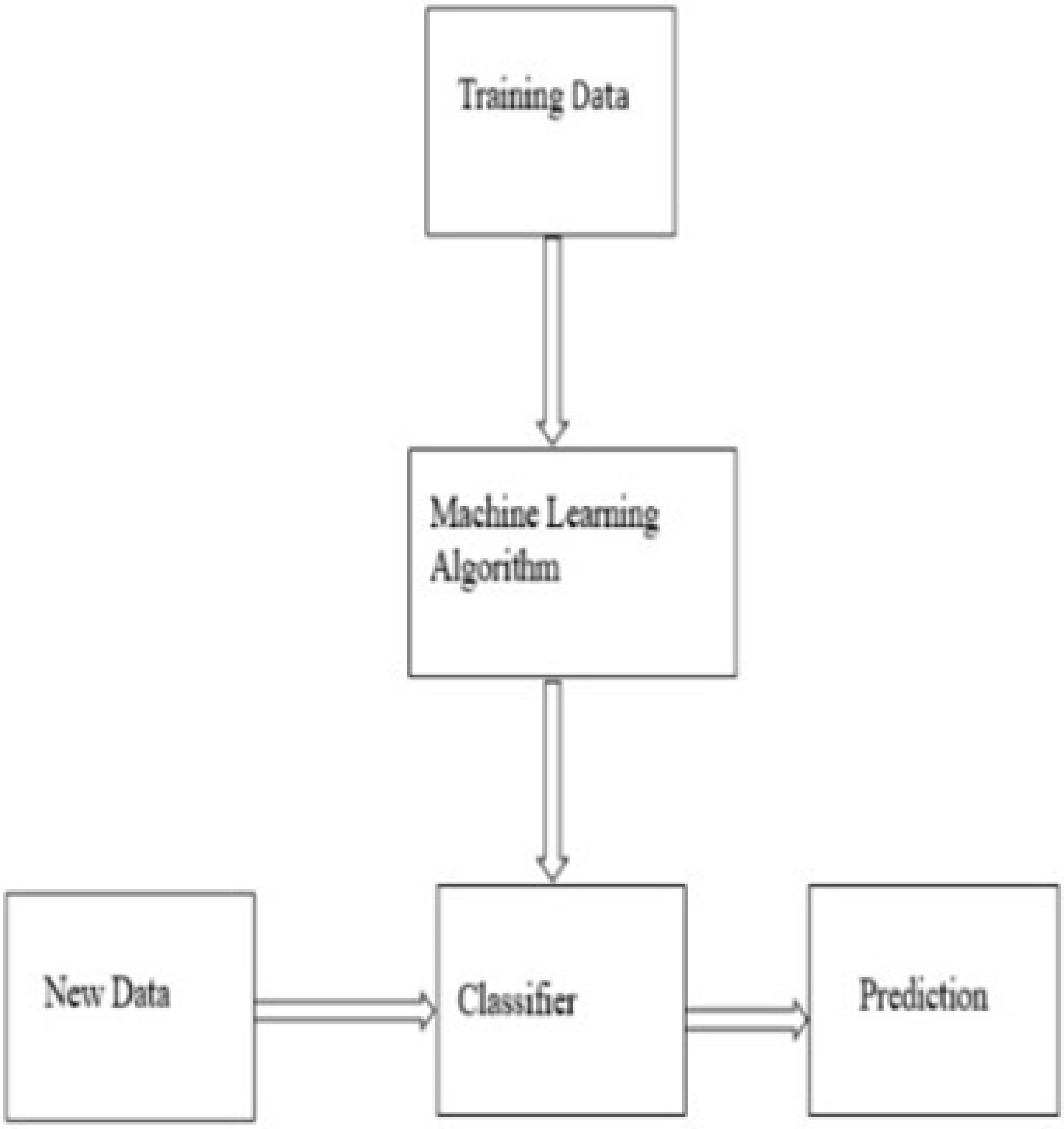


Fig 4.2.1: " System Flow Diagram"

4. REQUIREMENTS

Criterion	*Description*
OS Version	Windows 10 Home
RAM	8 GB RAM
DISK SPACE	1 TB
SCREEN RESOLUTION	1280 x 800

- Cython==0.29.13
- Flask==1.1.0
- Ipykernel==5.1.1
- Ipython==7.7.0
- Jupyter-client==5.3.1
- Jupyter-core==4.5.0
- Keras==2.2.4
- Keras-Applications==1.0.8
- Keras-Preprocessing==1.1.0
- Markdown==3.1.1
- Notebook==5.7.8
- Numpy==1.17.0
- Pandas==0.25.0
- Scikit-learn==0.21.2
- Tensorboard==1.14.0
- Tensorflow==1.14.0
- Tensorflow-estimator==1.14.0
- ATOM

Tool and Technology

5.1 Front end-

5.1.1Python

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 advantages-

1.Simple

Python is a simple and minimalistic language. Reading a good Python program feels almost like reading English, although very strict English! This pseudo-code nature of Python is one of its greatest strengths. It allows you to concentrate on the solution to the problem rather than the language itself.

2. Easy to Learn

As you will see, Python is extremely easy to get started with. Python has an extraordinarily simple syntax, as already mentioned.

3. Free and Open Source

Python is an example of a FLOSS (Open Source Software). In simple terms, you can freely distribute copies of this software, read it's source code, make changes to it, use pieces of it in new free programs, and that you know you can do these things. FLOSS is based on the concept of a community which shares knowledge. This is one of the reasons why Python is so good .it has been created and is constantly improved by a community who just want to see a better Python.

4.High-level Language

When you write programs in Python, you never need to bother about the low level details such as managing the memory used by your program, etc.

5. Portable

Due to its open-source nature, Python has been ported (i.e. changed to make it work on) to many platforms. All your Python programs can work on any of these platforms without requiring any changes at all if you are careful enough to avoid any system-dependent features. You can use Python on Linux, Windows, FreeBSD, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE and even PocketPC ! ￼

5. Interpreted

This requires a bit of explanation. A program written in a compiled language like C or C++ is converted from the source language i.e. C or C++ into a language that is spoken by your computer (binary code i.e. 0s and 1s) using a compiler with various fags a.3nd options.

6. Platform independent

The most important reason behind the popularity of **Python**, that it is a **Platform Independent Language**. Python is supported by many platforms like Linux, Windows, Macintosh etc.

Python is a High-Level programming language used everywhere to automate the boring stuff.

7. Large standard Library

Python’s standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as flee I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

The Python installers for the Windows platform usually include the entire standard library and often also include many additional components. For Unix-like operating systems Python is normally provided as a collection of packages, so it may be necessary to use the packaging tools provided with the operating system to obtain some or all of the optional components.

8. Expressive Language

It's a very expressive language with libraries for almost everything. It's used for web development, scientifc computing, desktop applications, etc. It's an easy language to learn but also powerful.

9. Object Oriented

Python supports procedure-oriented programming as well as object-oriented programming. In procedure-oriented languages, the program is built around procedures or functions which are nothing but reusable pieces of programs. In object-oriented languages, the program is built around objects which combine data and functionality. Python has a very powerful but simplistic way of doing OOP, especially when compared to big languages like C++ or Java.

5.1.1 Python Applications:

1. GUI-Based Desktop Applications:

Python has simple syntax, modular architecture, rich text processing tools and the ability to work on multiple operating systems which make it a desirable choice for developing desktop-based applications. There are various GUI toolkits like wxPython, PyQt or PyGtk available which help developers create highly functional Graphical User Interface (GUI). The various applications developed using Python includes:

2.Image Processing and Graphic Design Applications:

Python has been used to make 2D imaging software such as Inkscape, GIMP, Paint Shop Pro and Scribus. Further, 3D animation packages, like Blender, 3ds Max, Cinema 4D, Houdini, Lightwave and Maya, also use Python in variable proportions.

3.Scientific and Computational Applications:

The higher speeds, productivity and availability of tools, such as Scientific Python and Numeric Python, have resulted in Python becoming an integral part of applications involved in computation and processing of scientific data. 3D modeling software, such as Free CAD, and finite element method software, such as Abaqus, are coded in Python.

4. Games

Python has various modules, libraries and platforms that support development of games. For example, PySoy is a 3D game engine supporting Python 3, and PyGame provides functionality and a library for game development. There have been numerous games built using Python including Civilization-IV, Disney's Toontown Online, Vega Strike etc.

5. Web Frameworks and Web Applications:

Python has been used to create a variety of web-frameworks including CherryPy, Django, TurboGears, Bottle, Flask etc. These frameworks provide standard libraries and modules which simplify tasks related to content management, interaction with database and interfacing with different internet protocols such as HTTP, SMTP, XML-RPC, FTP and POP. Plone, a content management system; ERP5, an open source ERP which is used in aerospace, apparel and banking; Odoo – a consolidated suite of business applications; and Google App engine are a few of the popular web applications based on Python.

6. Enterprise and Business Applications:

With features that include special libraries, extensibility, scalability and easily readable syntax, Python is a suitable coding language for customizing larger applications. Reddit, which was originally written in Common Lips, was rewritten in Python in 2005. Python also contributed in a large part to functionality in YouTube.

7. Operating Systems:

Python is often an integral part of Linux distributions. For instance, Ubuntu's Ubiquity Installer, and Fedora's and Red Hat Enterprise Linux's Anaconda Installer are written in Python. Gentoo Linux makes use of Python for Portage, its package management system.

8. Language Development:

Python's design and module architecture has influenced development of numerous languages. Boo language uses an object model, syntax and indentation, similar to Python. Further, syntax of languages like Apple's Swift, CofeeScript, Cobra, and OCaml all share similarity with Python.

9.Prototyping:

Besides being quick and easy to learn, Python also has the open source advantage of being free with the support of a large community. This makes it the preferred choice for prototype development.

5.1.2

HTML

HTML

Hypertext Markup Language (**HTML**) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

Tags

HTML uses special bits of programming language called "tags" to let the browser know how a webpage should look. The tags *usually* come in pairs: an opening tag defines the start of a block of content and a closing tag defines the end of that block of content. There are many different kinds of tags, and each one has a different purpose. See **Basic HTML Tags** below for tag examples.

Some tags only work in certain browsers. For example, the <menu item> tag, which is used to make something appear when the person presses the right button of the mouse, only works on the Mozilla Firefox browser. Other browsers simply ignore this tag and display the writing normally. Many web page creators avoid using these "non-standard" tags because they want their pages to look the same with all browsers.

5.1. 3

Cascading Style Sheets(CSS)

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts.[3] This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS including XHTML, plain XML, SVG, and XUL.

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Bootstrap is the sixth-most-starred project on GitHub, with more than 135,000 stars, behind freeCodeCamp (almost 307,000 stars) and marginally behind Vue.js framework.According to Alexa Rank, Bootstrap is in the top-2000 in the USA while vuejs.org is in the top-7000 in the USA.

5.1.4

KERAS

Deep learning for humans.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

Iterate at the speed of thought.

Keras is the most used deep learning framework among top-5 winning teams on Kaggle. Because Keras makes it easier to run new experiments, it empowers you to try more ideas than your competition, faster. And this is how you win.

Exascale machine learning.

Built on top of TensorFlow 2.0, Keras is an industry-strength framework that can scale to large clusters of GPUs or an entire TPU pod. It's not only possible; it's easy.

Deploy anywhere.

Take advantage of the full deployment capabilities of the TensorFlow platform. You can export Keras models to JavaScript to run directly in the browser, to TF Lite to run on iOS, Android, and embedded devices. It's also easy to serve Keras models as via a web API.

A vast ecosystem.

Keras is a central part of the tighly-connected TensorFlow 2.0 ecosystem, covering every step of the machine learning workflow, from data management to hyperparameter training to deployment solutions.

State-of-the-art research.

Keras is used by CERN, NASA, NIH, and many more scientific organizations around the world (and yes, Keras is used at the LHC). Keras has the low-level flexibility to implement arbitrary research ideas while offering optional high-level convenience features to speed up experimentation cycles.

An accessible superpower.

Because of its ease-of-use and focus on user experience, Keras is the deep learning solution of choice for many university courses. It is widely recommended as one of the best ways to learn deep learning.

5.1.5

Pandas

In computer programming, **pandas** is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Library features

- DataFrame object for data manipulation with integrated indexing.
- Tools for reading and writing data between in-memory data structures and different file formats.

- Data alignment and integrated handling of missing data.
- Reshaping and pivoting of data sets.
- Label-based slicing, fancy indexing, and subsetting of large data sets.
- Data structure column insertion and deletion.
- Group by engine allowing split-apply-combine operations on data sets.
- Data set merging and joining.
- Hierarchical axis indexing to work with high-dimensional data in a lower-dimensional data structure.
- Time series-functionality: Date range generation[4] and frequency conversion, moving window statistics, moving window linear regressions, date shifting and lagging.
- Provides data filtration.

The library is highly optimized for performance, with critical code paths written in Cython or C

5.1.6

TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging .

Robust ML production anywhere

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

TensorFlow offers multiple levels of abstraction so you can choose the right one for your needs. Build and train models by using the high-level Keras API, which makes getting started with TensorFlow and machine learning easy.

If you need more flexibility, eager execution allows for immediate iteration and intuitive debugging. For large ML training tasks, use the Distribution Strategy API for distributed training on different hardware configurations without changing the model definition.

5.1.7

Anaconda

Anaconda is a free and open source distribution of the Python and R programming languages for data science and machine Learning. Package versions are managed by the package management system conda. The Anaconda distribution is used by over 6 million users, and it includes more than 250 popular data science packages suitable for Windows, Linux, and MacOS.

Anaconda distribution comes with more than 1,000 data packages as well as the Conda package and virtual environment manager, called **Anaconda Navigator** , so it eliminates the need to learn to install each library independently.

Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

Navigator is automatically installed when you install Anaconda version 4.0.0 or higher.

The following applications are available by default in Navigator:

1. JupyterLab
2. Jupyter Notebook
3. QTConsole
4. Spyder
5. VSCode
6. Glueviz
7. Orange 3 App
8. Rodeo
9. RStudio

5.2 Backend-

5.2.1

Flask

Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries.[3] It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far

more frequently than the core Flask program.[4]

Applications that use the Flask framework include Pinterest and LinkedIn.

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

Flask offers suggestions, but doesn't enforce any dependencies or project layout. It is up to the developer to choose the tools and libraries they want to use. There are many extensions provided by the community that make adding new functionality easy.

5.2.2

MACHINE LEARNING

Machine learning (**ML**) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis through unsupervised learning. In its application across business problems, machine learning is also referred to as predictive analytics.

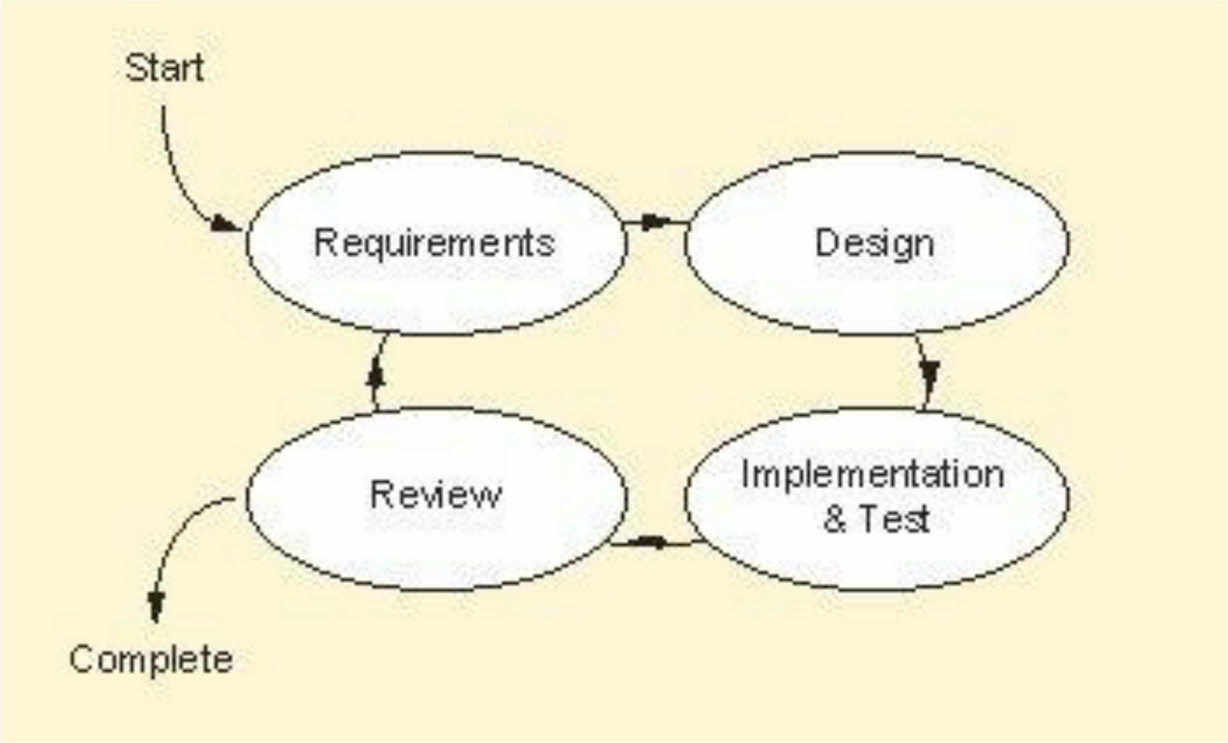
ML concepts used in project are :

- Neural network
- Long Short Term Memory(LSTM)
- Sigmoid Function
- Regularization

MODEL USED

6.1 Iterative Model

An iterative lifecycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which can then be reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software for each cycle of the model. Consider an iterative lifecycle model which consists of repeating the following four phases in sequence:



1. A ***Requirements*** phase, in which the requirements for the software are gathered and analyzed. Iteration should eventually result in a requirements phase that produces a complete and final specification of requirements.

2. A ***Design*** phase, in which a software solution to meet the requirements is designed. This may be a new design, or an extension of an earlier design.

3. An ***Implementation and Test*** phase, when the software is coded, integrated and tested.

4. A ***Review*** phase, in which the software is evaluated, the current requirements are reviewed, and changes and additions to requirements proposed.

5. For each cycle of the model, a decision has to be made as to whether the software produced by the cycle will be discarded, or kept as a starting point for the next cycle (sometimes referred to as incremental prototyping). Eventually a point will be reached where the requirements are complete and the software can be delivered, or it becomes impossible to enhance the software as required, and a fresh start has to be made.

6. The iterative lifecycle model can be likened to producing software by successive approximation. Drawing an analogy with mathematical methods that use successive

approximation to arrive at a final solution, the benefit of such methods depends on how rapidly they converge on a solution.

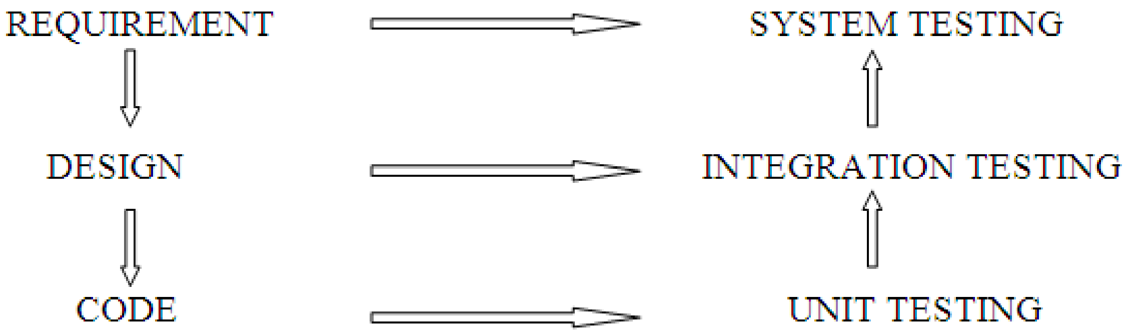
7. The key to successful use of an iterative software development lifecycle is rigorous validation of requirements, and verification (including testing) of each version of the software against those requirements within each cycle of the model. The first three phases of the example iterative model is in fact an abbreviated form of a sequential V or waterfall lifecycle model. Each cycle of the model produces software that requires testing at the unit level, for software integration, for system integration and for acceptance. As the software evolves through successive cycles, tests have to be repeated and extended to verify each version of the software.

TESTING

Testing is the major quality control measure employed during software development. **Testing** is the process of executing the program with the intent of finding an error. No piece of code is completely ready unless it has been fully tested. This stage is very important. At this stage it is verified whether the code developed meet the requirement specification or not. However, all validation is also checked in the testing stage.

Our project includes the following types of testing:

1. Unit testing.
2. Integration testing.
3. System testing.



6.1 UNIT TESTING

The Unit testing is the first level of testing in our project. In this testing, we have tested the different modules against the specification produced for the modules. Unit testing is essential for the verification of the code produced during the coding phase and hence the goal is the internal logic of the modules.

6.2 INTEGRATION TESTING

In this, many tested modules are combined into subsystem, which are then tested. The goal here is to see if the modules can be integrated properly, the emphasis be considered as testing the design, and hence the emphasis on testing module interaction.

6.3 SYSTEM TESTING

Here the entire software system is tested. The reference document for this process is requirement document, and the goal is to see if the software meets its requirements. This is essentially a validation exercise.

6.4 VERIFICATION AND VALIDATION

Verification is the process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

"Are we building the product right?"

Validation is the process of evaluating software during or at the end of development process to determine whether it satisfies specified condition.

"Are we building the right product?"

Verification and validation checks of our project

- **No field cannot be empty:** With this check, no field is to be entered blank or empty. If user will do this then the message will be displayed that this field is compulsory.
- **Password has minimum number of characters:** According to this feature the user has to set his/her password with minimum 6 digits/characters***.***

BIBLIOGRAPHY:

To bring the system to verge of completion the following books have been referred:

NAME OF THE BOOK	AUTHOR'S NAME
Learning Python	O'Rielly
Python Programming	John M.Zelle
Hands on Machine Learning with Scikit-learn & Tensorflow	Aurelien Geron

Some websites referred are:-

- https://www.w3schools.com
- https://www.tutorialspoint.com/
- www.keras. Com
- www.tensorflow.com
- www.ieee.com

ORGANIZATION PROFILE

There is a reason why we take pride in branding ourselves as the best programming language platform in India. Our instructors, graduates from Stanford University, IITs, IIITs, are Master craftsmen with years of industry experience at Facebook, Amazon, American Express, Times Internet, etc. At Coding Ninjas, our mission is to continuously innovate the best ways to train the next generation of developers and to transform the way tech education is delivered. We're constantly evolving how we train amazing developers, as staying stagnant is not an option. We approach our educational philosophy as a never-ending journey of self-improvement and we apply it to everything we do Some of the courses offered are

- Machine Learning
- Python with data structure
- C++ with data structure
- Java with data structure
- Data Science
- Web application development
- Android app development