

A
Project Report
On
Face And Eye Detection Using Python And Artificial Intelligence

A Dissertation Submitted to Jawaharlal Nehru Technological University Hyderabad in partial fulfillment
of the requirements for the award of the degree of

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

| | |
|--------------------------------|-------------------|
| Abhishek Banda | 167W1A0597 |
| Duggu Harish Kumar | 167W1A0598 |
| Ankit | 167W1A05A0 |
| Ejenthkar Amarnath Ajay | 157W1A0521 |
| Katakam Nagaraju | 157W1A0535 |

Under the Esteemed Guidance of

T.Nagarjun

Assistant Professor



(2016-2020)

ST. MARY'S INTEGRATED CAMPUS HYDERABAD

(Approved by AICTE, Permitted by Govt. of Telangana, Affiliated to JNTUH)
Behind Mount Opera, Deshmukhi(V), Pochampalli(M), Yadadri(Dt), Telangana, 508284.



St. Mary's Integrated Campus Hyderabad

(Approved by AICTE, Accredited by NAAC, Permitted by Govt. of Telangana, Affiliated to JNTU Hyderabad)

Near Ramoji Film City, Behind Mount Opera, Deshmukhi(V), Pochampally(M), Yadadri Bhongir(Dt.), & Satasingaram(V), Hayathnagar

(M), Rangareddy(Dt.), Greater Hyderabad -508 284. Under HMDA T.S., INDIA. Mobile: 8498096423, 8498096465

Phone: 040-65799096, Fax: 040-66809093. Website: smichyderabad.org E-mail: principal_smic@rediffmail.com

CERTIFICATE

This is to certify that the project work entitled “**Face And Eye Detection Using Python And Artificial Intelligence**” is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering** from **Jawaharlal Nehru Technological University Hyderabad** is a bonafide work of

| S.No | Hall Ticket No | Name Of The Student |
|------|----------------|-------------------------|
| 1. | 167W1A0597 | Abhishek Banda |
| 2. | 167W1A0598 | Duggu Harish Kumar |
| 3. | 167W1A05A0 | Ankit |
| 4. | 157W1A0521 | Ejenthkar Amarnath Ajay |
| 5 | 157W1A0535 | Katakam Nagaraju |

Head Of The Department

Dr. M. Malla Reddy

Principal

Dr. E. Swathin Goud

Internal Guide

T. Nagarjun

Assistant Professor

External Examiner

Sponsored by : **Joseph Sriharsha & Mary Indrara Educational Society**

Regd. Off : Plot No. 102, High Court Colony, Vanasthalipuram, Hyderabad – 500 070. T.S., INDIA.

Corp. Off : 3rd Floor, Rajnigandha Apts., Opp.: Chaitanyapuri Bus Stop, Dilukhnagar, Hyderabad – 500 060. T.S., INDIA.

ACKNOWLEDGEMENT

This is an acknowledgement of the intensive drive and technical competence of many individuals who have contributed to the success of my project.

We are sincerely thankful to our Principal, **Dr. E. Swathin Goud**, St. Mary's Integrated Campus Hyderabad and to all my faculty members.

We are grateful to Director, **Mr. G. Satyanarayana Reddy**, St. Mary's Integrated Campus Hyderabad for granting me the permission for undergoing the practical training through development of this project.

We are immensely thankful to my internal guide **T. Nagarjun**, Assistant Professor of C.S.E, for his valuable guidance and suggestions in each and every stage of this work, which helped me in completing this project work successfully.

We are thankful to our parents, faculties and all who co-operated us to complete our project successfully.

| | |
|-------------------------|--------------|
| Banda Abhishek | - 167W1A0597 |
| Duggu Harish Kumar | - 167W1A0598 |
| Ankit | - 167W1A05A0 |
| Ejenthkar Amarnath Ajay | - 157W1A0521 |
| Katakam Nagaraju | - 157W1A0535 |

DECLARATION

We declare that the project work reported in the entitled “**Face And Eye Detection Using Python And Artificial Intelligence**” a record of the work done by me in the **Department of Computer Science and Engineering**, St. Mary’s Integrated Campus, Hyderabad.

No part of the thesis is copied from books/journals/Internet and where ever referred, the same has been duly acknowledged in the text. The reported data are based on the project work done entirely by me/we and not copied from any other source.

Abhishek Banda - 167W1A0597

Duggu Harish Kumar - 167W1A0598

Ankit - 167W1A05A0

Ejenthkar Amarnath Ajay - 157W1A0521

Katakam Nagaraju - 157W1A0535

Place: Hyderabad

Date:

ABSTRACT

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene.

Face detection can be regarded as a specific case of object-class detection. In object-class detection, the task is to find the locations and sizes of all objects in an image that belong to a given class. Examples include upper torsos, pedestrians, and cars.

Face-detection algorithms focus on the detection of frontal human faces. It is analogous to image detection in which the image of a person is matched bit by bit. Image matches with the image stores in database. Any facial feature changes in the database will invalidate the matching process. A reliable face-detection approach based on the genetic algorithm and the eigen-face technique.

Firstly, the possible human eye regions are detected by testing all the valley regions in the gray-level image. Then the genetic algorithm is used to generate all the possible face regions which include the eyebrows, the iris, the nostril and the mouth corners.

Each possible face candidate is normalized to reduce both the lighting effect, which is caused by uneven illumination; and the shirring effect, which is due to head movement. The fitness value of each candidate is measured based on its projection on the eigen-faces. After a number of iterations, all the face candidates with a high fitness value are selected for further verification. At this stage, the face symmetry is measured and the existence of the different facial features is verified for each face candidate.

Face detection is gaining the interest of marketers. A webcam can be integrated into a television and detect any face that walks by. The system then calculates the race, gender, and age range of the face. Once the information is collected, a series of advertisements can be played that is specific toward the detected race/gender/age.

CONTENTS

| CHAPT ER NO | TOPIC NO | TITLE | PAGE NO |
|----------------|-------------|---------------------------------------|-----------|
| | | ACKNOWLEDGEMENT | i |
| | | DECLARATION | ii |
| | | ABSTRACT | iii |
| | | LIST OF CONTENTS | iv |
| | | LIST OF FIGURES | vi |
| 1 | 1 | INTRODUCTION | 1 |
| | 1.1 | Contributions | 2 |
| 2 | 2 | SYSTEM ANALYSIS | 3 |
| | 2.1 | Definition | 4 |
| | 2.2 | Description of Present System | 4 |
| | 2.2.1 | Limitations of Present System | 4 |
| | 2.3 | Proposed System | 4 |
| | 2.3.1 | Advantages | 4 |
| | 2.4 | Feasibility Study | 5 |
| | 2.4.1 | Technical Feasibility | 5 |
| | 2.4.2 | Economic Feasibility | 5 |
| | 2.4.3 | Behavioural / Operational Feasibility | 6 |
| 3 | 3 | SYSTEM SPECIFICATIONS | 7 |
| | 3.1 | Hardware Specifications | 8 |
| | 3.2 | Software Specifications | 8 |
| 4 | 4 | SYSTEM DESIGN | 9 |
| | 4.1 | Definition | 10 |
| | 4.2 | Output Design | 10 |
| | 4.3 | Input Design | 10 |
| | 4.4 | Logical Design | 11 |
| 5 | 5 | IMPLEMENTATION | 12 |
| | 5.1 | OpenCV | 13 |
| | 5.2 | Haar Cascade Algorithm | 13 |
| 6 | 6 | SOFTWARE ENVIRONMENT | 15 |
| | 6.1 | Python | 16 |
| | 6.2 | Java Vs. Python In 2020 | 16 |
| | 6.3 | Java Vs. Python Typing | 17 |
| | 6.4 | Code Readability and Formatting | 18 |
| | 6.5 | Whitespace | 18 |
| | 6.6 | Brevity | 18 |
| | 6.7 | Performance | 19 |
| | 6.8 | Final Thoughts on Java Vs. Python | 20 |

| | | | |
|-----------|-----------|--|-----------|
| | 6.9 | What Is Machine Learning? | 23 |
| | 6.9.1 | Machine Learning Vs. Traditional Programming | 23 |
| | 6.10 | How Does Machine Learning Work? | 23 |
| | 6.11 | Machine Learning Algorithms and Where They Are Used? | 25 |
| | 6.12 | Classification | 25 |
| | 6.12.1 | Supervised Learning | 26 |
| | 6.12.2 | Unsupervised Learning | 27 |
| | 6.12.3 | Reinforcement Learning | 28 |
| | 6.13 | How to Choose Machine Learning Algorithm | 32 |
| | 6.14 | Challenges and Limitations of Machine Learning | 32 |
| | 6.15 | Application of Machine Learning | 33 |
| | 6.16 | Why Is Machine Learning Important? | 35 |
| 7 | 7 | SAMPLE CODE | 36 |
| 8 | 8 | SYSTEM TESTING | 39 |
| | 8.1 | Unit Testing | 40 |
| | 8.2 | Integration Testing | 41 |
| | 8.3 | Validation Testing | 42 |
| | 8.4 | High-Order Testing | 42 |
| | 8.5 | Bottom-Up Integration Testing | 42 |
| | 8.6 | Regression Testing | 42 |
| 9 | 9 | SCREEN SHOTS | 43 |
| 10 | 10 | CONCLUSION | 48 |
| 11 | 11 | FUTURE ENHANCEMENT | 50 |
| 12 | 12 | BIBLIOGRAPHY | 52 |
| | 12.1 | References | 53 |

LIST OF FIGURES

| TOPIC NO | FIGURE NO | FIGURE NAME | PAGE NO |
|------------|-----------|---------------------------------|-----------|
| 6.1 | 1 | Machine Learning | 13 |
| 6.2 | 2 | Learning Phase | 14 |
| 6.3 | 3 | Machine Learning Implementation | 15 |
| 6.4 | 4 | Substructure | 16 |
| 6.5 | 5 | RI Scenario | 26 |
| 6.6 | 6 | Average salaries | 27 |
| 9.1 | 7 | Anaconda Navigator | 42 |
| 9.2 | 8 | Anaconda Prompt | 43 |
| 9.3 | 9 | Turning on Spyder | 44 |
| 9.4 | 10 | Running Code in Spyder | 45 |
| 9.5 | 11 | Files in Spyder | 45 |
| 9.6 | 12 | Face And Eyes Found | 46 |

CHAPTER-1

INTRODUCTION

1.INTRODUCTION

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images.Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene.

Face detection is gaining the interest of marketers. A webcam can be integrated into a television and detect any face that walks by. The system then calculates the race, gender, and age range of the face. Once the information is collected, a series of advertisements can be played that is specific toward the detected race/gender/age.An example of such a system is OptimEyes and is integrated into the Amscreen digital signage system

Some recent digital cameras use face detection for autofocus. Face detection is also useful for selecting regions of interest in photo slideshows that use a pan-and-scale Ken Burns effect.Modern appliances also use smile detection to take a photograph at an appropriate time.

Face detection is used in biometrics, often as a part of (or together with) a facial recognition system. It is also used in video surveillance, human computer interface and image database management.

Face detection can be used as part of a software implementation of emotional inference. Emotional inference can be used to help people with autism under stand the feelings of people around them.

Face detection is essential for the process of language inference from visual queues. Lip reading has applications in help computers determine who is speaking which is needed when security is important.

CHAPTER-2

SYSTEM ANALYSIS

2.SYSTEM ANALYSIS

2.1 DEFINITION

System Analysis is the detailed study of the various operations performed by the system and their relationships within and outside the system. Analysis is the process of breaking something into its parts so that the whole may be understood. System analysis is concerned with becoming aware of the problem, identifying the relevant and most decisional variables, analyzing and synthesizing the various factors and determining an optional or at least a satisfactory solution. During this a problem is identified, alternate system solutions are studied and recommendations are made about committing the resources used to the system.

2.2 DESCRIPTION OF PRESENT SYSTEM

Face detection is gaining the interest of marketers. A webcam can be integrated into a television and detect any face that walks by. The system then calculates the race, gender, and age range of the face. Once the information is collected, a series of advertisements can be played that is specific toward the detected race/gender/age. An example of such a system is OptimEyes and is integrated into the Amscreen digital signage system which is not achieved in Present System

2.2.1 LIMITATIONS OF PRESENT SYSTEM

Analysis is done manually. These tasks are time consuming. It may take one month or long. Analysis using machine learning is very convenient because in the manual system there are lot of difficulties in conducting.

2.3 PROPOSED SYSTEM

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection also refers to the psychological process by which humans locate and attend to faces in a visual scene.

2.3.1 ADVANTAGES

Face detection can be used as part of a software implementation of emotional inference. Emotional inference can be used to help people with autism understand the feelings of people around them.

The aim is to automate its existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements.

2.4 FEASIBILITY STUDY

A feasibility analysis usually involves a thorough assessment of the operational (need), financial and technical aspects of a proposal. Feasibility study is the test of the system proposal made to identify whether the user needs may be satisfied using the current software and hardware technologies, whether the system will be cost effective from a business point of view and whether it can be developed with the given budgetary constraints. A feasibility study should be relatively cheap and done at the earliest possible time. Depending on the study, the decision is made whether to go head with a more detailed analysis.

When a new project is proposed, it normally goes through feasibility assessment. Feasibility study is carried out to determine whether the proposed system is possible to develop with available resources and what should be the cost consideration. Facts considered in the feasibility analysis were

Technical Feasibility

Economic Feasibility

Behavioral Feasibility

2.4.1 Technical Feasibility

Technical feasibility includes whether the technology is available in the market for development and its availability. The assessment of technical feasibility must be based on an outline design of system requirements in terms of input, output, files, programs and procedures. This can be qualified in terms of volumes of data, trends, frequency of updating, cycles of activity etc, in order to give an introduction of technical system. Considering our project it is technical feasible. Gender classification Analysis and Gender classification Analysis Systems,

with its emphasis on a more strategic decision making process is fast gaining ground as a popular outsourced function.

2.4.2 Economic Feasibility

This feasibility study presents tangible and intangible benefits from the project by comparing the development and operational cost. The technique of cost benefit analysis is often used as a basis for assessing economic feasibility. This system needs some more initial investment than the existing system, but it can be justifiable that it will improve quality of service.

Thus feasibility study should center along the following points:

- Improvement resulting over the existing method in terms of accuracy, timeliness.

- Cost comparison

- Estimate on the life expectancy of the hardware.

- Overall objective.

Our project is economically feasible. It does not require much cost to be involved in the overall process. The overall objective is in easing out the Gender classification Analysis processes.

2.4.3 Behavioral / Operational Feasibility

This analysis involves how it will work when it is installed and the assessment of political and managerial environment in which it is implemented. People are inherently resistant to change and computers have been known to facilitate change. The new proposed system is very much useful to the users and therefore it will accept broad audience from around the world.

CHAPTER-3

SYSTEM SPECIFICATIONS

3.SYSTEM SPECIFICATIONS

3.1 Hardware Specification

| | | |
|-----------|---|-------------------------|
| CPU | : | CORE I3/I5/I7 PROCESSOR |
| SPEED | : | 2 GHz COPROCESSOR |
| TOTAL RAM | : | 8 GB |
| HARD DISK | : | 500 GB |
| KEYBOARD | : | 105 KEYS |
| MOUSE | : | LOGITECH MOUSE |
| DISPLAY | : | SGVA COLOR |

3.2 Software Specification

| | | |
|----------------------|---|----------------------|
| TECHNOLOGY | : | MACHINE LEARNING |
| PROGRAMMING LANGUAGE | : | PYTHON |
| DISTRIBUTION | : | ANACONDA |
| OPERATING SYSTEM | : | WINDOWS 8/WINDOWS 10 |

CHAPTER-4

SYSTEM DESIGN

4.SYSTEM DESIGN

4.1 DEFINITION

The most creative and challenging face of the system development is System Design. It provides the understanding and procedural details necessary for the logical and physical stages of development. In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfill. The first step is to determine how the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing.

Design of the system can be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus system design is a solution to “how to” approach to the creation of a new system. This important phase provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The design step provides a data design, architectural design, and a procedural design.

4.2 OUTPUT DESIGN

In the output design, the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printers and the screen. Most users now access their reports from either a hard copy or screen display. Computer’s output is the most important and direct source of information to the user, efficient, logical, output design should improve the systems relations with the user and help in decision-making.

As the outputs are the most important source of information to the user, better design should improve the systems relations and also should help in decision-making. The output device’s capability, print quality, response time requirements etc should also be considered, form design elaborates the way the output is presented and layout available for capturing information. It’s very helpful to produce the clear, accurate and speedy information for end users.

4.3 INPUT DESIGN

In the input design, user-originated inputs are converted into a computer-based system format. It also includes determining the record media, method of input, speed of capture and entry on to the screen. Online data entry accepts commands and data through a keyboard. The major approach to input design is the menu and the prompt design. In each alternative, the user's options are predefined. The data flow diagram indicates logical data flow, data stores, source and destination. Input data are collected and organized into a group of similar data once identified input media are selected for processing.

In this software, importance is given to develop Graphical User Interface (GUI), which is an important factor in developing efficient and user friendly software. For inputting user data, attractive forms are designed. User can also select the desired options from the menu, which provides all possible facilities. Also the important input format is designed in such a way that accidental errors are avoided. The user has to input only just the minimum data required, which also helps in avoiding the errors that the users may make. Accurate designing of the input format is very important in developing efficient software. The goal of input design is to make entry as easy, logical and free from errors.

4.4 LOGICAL DESIGN

Logical data design is about the logically implied data. Each and every data in the form can be designed in such a manner to understand the meaning. Logical data designing should give a clear understanding & idea about the related data used to construct a form.

CHAPTER-5

IMPLEMENTATION

5.IMPLEMENTATION

5.1 OpenCV

Open source computer vision library is an open source computer vision and machine learning library. It was built with a vision to provide basic infrastructure to the computer vision application.

5.2 Haar Cascade Algorithm

It is a machine learning algorithm used to identify objects in image or video based on the concepts of features proposed by Paul Viola and Michael Jones in 2001.

The algorithm contains four stages:

Haar Feature Selection

Creating Intergal Images

Adaboost Training

Cascading Classifiers

Before jumping into the code you have to install OpenCV into your Odinub. You can check out the steps from [here](#).

Step 1: Create a new Python file using the following command:

```
gedit filename.py
```

Step 2: Now before starting the code import the modules of OpenCV as following:

The following command will enable the code to do all the scientific computing. It will enable the code to carry out different operations:

```
import numpy as np
```

The following module will make available all the functionalities of the OpenCV library.

```
import cv2
```

Step 3: The cascade classifiers

The cascade classifiers are the trained.xml files for detecting the face and eyes.

```
face_cascade=cv2.CascadeClassifier('/root/opencv/data/haarcascades/haarcascade_frontalface_default.xml')
eye_cascade=cv2.CascadeClassifier('root/opencv/data/haarcascades/haarcascade_eye.xml')
```

Step 4: The imread() function

The imread() function is used to read the image captured by passing the path of the image as the input parameter in form of string.

```
img=cv2.imread('/root/Desktop/baby.jpg').
```

Step 5: The cvtColor() function

It converts the image from one color space to another.

```
cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Step 6: The imshow() function

It is used to display the image on the window. Here the first command is the string which will assign the name to the window. The second argument is the image that is to be displayed into the window.

```
cv2.imshow('img',img)
```

Step 7: The waitKey() function

It will wait generate delay for the specified milliseconds.

```
cv2.waitKey()
```

Step 8: The destroyAllWindows() function

This function will destroy all the previously created windows.

```
cv2.destroyAllWindows
```

Step 9: Simply run your code with the help of following command

```
python filename.py
```

CHAPTER-6

SOFTWARE ENVIRONMENT

6.SOFTWARE ENVIRONMENT

6.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

6.2 Java vs. Python in 2019

Java and Python have many similarities. Both languages have strong cross-platform support and extensive standard libraries. They both treat (nearly) everything as objects. Both languages compile to bytecode, but Python is (usually) compiled at runtime. They are both members of the Algol family, although Python deviates further from C/C++ than Java does.

Support for Python 2.x will end on January 1, 2020. For a long time, Python development has fragmented between version 2.7 and the regular releases of new 3.x versions. But, with the end- of-life date for Python 2 a year away, the question over which version to use is settled. The community has centered on Python 3.

Meanwhile, [Oracle's new release model for Java](https://blogs.oracle.com/java-platform-group/update-and-faq-on-the-java-se-release-cadence) [HYPERLINK](https://blogs.oracle.com/java-platform-group/update-and-faq-on-the-java-se-release-cadence)
"https://blogs.oracle.com/java-platform-group/update-and-faq-on-the-java-se-release-cadence"

" [HYPERLINK](#)

"<https://blogs.oracle.com/java-platform-group/update-and-faq-on-the-java-se-release-cadence>

" [HYPERLINK](#)

"<https://blogs.oracle.com/java-platform-group/update-and-faq-on-the-java-se-release-cadence>

" [HYPERLINK](#) created a lot of fear, uncertainty, and doubt in the software community. Even though the announcement provided a free (as in beer) option and a clear upgrade path, [confusion continues to reign](#). Several platforms providers, such as Red Hat and Amazon, have stepped in to support OpenJDK. But the once unified Java community is more fragmented than Python ever was. Let's take a closer look at the similarities and differences between Java vs. Python.

6.3 Java vs. Python typing

Python and Java are both object-oriented languages, but Java uses static types, while Python is dynamic. This is the most significant difference and affects how you design, write, and troubleshoot programs in a fundamental way. Let's look at two code examples.

First, in Python, we'll create an array with some data in it, and print it to the console.

Next, let's try it in Java.

In Python, we put two strings and an integer in the same array, and then printed the contents. For Java, we declared a List of Strings and put three string values in it.

We can't mix types in a Java array. The code won't compile.

```
String array[] = {"Hello, World", "Hi there, Everyone", 6};
```

We could declare the array as containing Object instead of String, and override Java's type system. But, that's not how any experienced Java developer uses the language.

In Python, we don't have to provide a type when we declare the array and can put whatever we want in it. It's up to us to make sure we don't try to misuse the contents.

For example, what if we modified the code above to do this?

The above code will throw an error when we try to run it since we can't append an integer with a string. What are advantages and disadvantages to dynamic typing and static typing?

Static typing catches type errors at compile time. So, if mixing strings and integers weren't what you wanted to do, the Java compiler catches the mistake. How much of an advantage compile-time checks is up for debate in some circles. But, static typing does enforce a discipline that some developers appreciate.

Whether static typing prevents errors or not, it does make code run faster. A compiler working on statically-typed code can optimize better for the target platform. Also, you avoid runtime type errors, adding another performance boost.

Code that's written with dynamic types tends to be less verbose than static languages. Variables aren't declared with types, and the type can change. This saves a copy or type conversion to new variable declarations.

The question of code readability comes up often when debating Java vs. Python. Let's take a look at that next.

6.4 Code readability and formatting

Let's take an example from Java and Python and compare them. In this example, we need to open a large text file and collect each line into sets of 50 comma-separated records. Here is the Python code:

Here's the Java code:

6.5 Whitespace

Whitespace is part of Python's syntax, while Java ignores it. Python uses tabs for nesting and a full colon to start loops and conditional blocks. Java ignores whitespace and uses semicolons, parentheses and curly braces. Arguments over which code is easier to read, like the debate over static vs. dynamic typing, are subjective. Some say Python code is more concise and uniform than Java because your formatting choices are more limited. Python's use of whitespace ends debates over how to format code. The only option you have left is how to use blank lines.

The Python snippet is a few lines shorter than the Java snippet, a difference that adds up in larger programs. Much of the difference is because there are no closing braces. But Python's brevity—when compared to Java—goes deeper.

6.6 Brevity

Let's look at how the two languages handle files. Here's Python again:

```
with open(file_name, "r") as in_file:
```

Here's Java:

```
try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {
```

In both cases, the declaration creates a block. The file resource remains in scope, and the languages close it when the code exits the block.

In Python, we're opening a file and reading from it. When the loop reaches the end of the file, the loop exits.

```
for line in in_file:
```

Java is more complicated. We're opening a `BufferedReader` by passing it a `FileReader`. We consume lines from the reader. It's our responsibility to check for null when the file ends.

```
while ((line = reader.readLine()) != null) {
```

This only demonstrates that it's easier to handle text files in Python. But, it demonstrates how Java tends to be more verbose than Python. "Pythonic" constructs are more concise and less demanding. Java has evolved over the past few releases, with the introduction of try-with-resources in Java 7 and lambdas in Java 8, but it's still a verbose language.

Let's revisit our first example. Here's the Python again:

Here is the Java:

Both of these snippets will build and run as is. Python will run a script from beginning to end of a file. Java requires at least one entry point, a static method named `main`. The JVM (Java virtual machine) runs this method in the class passed to it on the command line.

Putting together a Python program tends to be faster and easier than in Java. This is especially true of utility programs for manipulating files or retrieving data from web resources.

6.7 Performance

Both Java and Python compile to bytecode and run in virtual machines. This isolates code from differences between operating systems, making the languages cross-platform. But there's a critical difference. Python usually compiles code at runtime, while Java compiles it in advance, and distributes the bytecode.

Most JVMs perform just-in-time compilation to all or part of programs to native code, which significantly improves performance. Mainstream Python doesn't do this, but a few variants such as [PyPy](http://pypy.org/) [HYPERLINK "http://pypy.org/"](http://pypy.org/) [HYPERLINK "http://pypy.org/"](http://pypy.org/) [HYPERLINK "http://pypy.org/"](http://pypy.org/) [HYPERLINK "http://pypy.org/"](http://pypy.org/)

The difference in performance between Java and Python is [sometimes significant](#) [HYPERLINK](#) _some cases. A simple binary tree test runs ten times faster in Java than in Python.

6.8 Final thoughts on Java vs. Python

So, which language is your best choice?

Oracle's new support model changes the Java landscape. While there is still a free option, the new release schedule and support model gives developers a reason to take stock. Java clients will need to pay Oracle for support, change OpenJDK versions on a regular basis, or rely on third parties like Red Hat or Amazon for fixes and security updates.

At the same time, Python has cleared a significant hurdle with Python 3. Python has a more unified support model than Java for the first time, and open source developers are focusing their efforts on the latest version of the language. I have to give Python the edge here.

Whether Python's dynamic typing is better than Java's static approach is subjective. The debate between the two models predates both of them, and it's a question of what's best for you and your team.

After working on large projects in both languages, I feel secure saying that Python's syntax is more concise than Java's. It's easier to get up and running quickly with a new project in Python than it is in Java. Python wins again.

Performance is where Java has a substantial advantage over Python. Java's just-in-time compilation gives it an advantage over Python's interpreted performance. While neither language is suitable for latency-sensitive applications, Java is still a great deal faster than Python.

All things considered, Python's advantages outweigh the disadvantages. If you're not already considering it, give it another look.

What is it?

Java is a general-purpose object-oriented programming language used mostly for developing a wide range of applications from mobile to web to enterprise apps.

Python is a high-level object-oriented programming language used mostly for web development, artificial intelligence, machine learning, automation, and other data science applications.

Creator

Java was created by James Gosling (Sun Microsystems).

Python was created by Guido van Rossum.

Open source status

Java is free and (mostly) open source except for corporate use.

Python is free and open source for all use cases.

Platform dependencies

Java is platform-independent (although JVM isn't) per its WORA ("write once, run anywhere") philosophy.

Python is platform-dependent.

Compiled or interpreted

Java is a compiled language. Java programs are translated to byte code at compile time and not runtime.

Python is an interpreted language. Python programs are translated at runtime.

File creation

Java: After compilation, **<filename>.class** is generated.

Python: During runtime, **<filename>.pyc** is created.

Errors types

Java has 2 types of errors: compile and runtime errors.

Python has 1 error type: traceback (or runtime) error.

Statically or dynamically typed

Java is statically typed. When initiating variables, their types need to be specified in the program because type checking is done at compile time.

Python is dynamically typed. Variables don't need to have a type specified when initiated because type checking is done at runtime.

Syntax

Java: Every statement needs to end with a semicolon (;), and blocks of code are separated by curly braces ({ }).

Python: Blocks of code are separated by indentation (the user can choose how many white spaces to use, but it should be consistent throughout the block).

Number of classes

Java: Only one public top-level class can exist in a single file in Java.

Python: Any number of classes can exist in a single file in Python.

More or less code?

Java generally involves writing more lines of code compared to Python.

Python involves writing fewer lines of code compared to Java.

Multiple inheritance

Java does not support multiple inheritance (inheriting from two or more base classes)

Python supports multiple inheritance although it is rarely implemented due to various issues like inheritance complexity, hierarchy, dependency issues, etc.

Multi-threading

Java multi-threading can support two or more concurrent threads running at the same time.

Python uses a global interpreter lock (GIL), allowing only a single thread (CPU core) to run at a time.

Execution speed

Java is usually faster in execution time than Python.

Python is usually slower in execution time than Java.

Hello world in Java

```
public class Hello {  
  
    public static void main(String[] args) { System.out.println("Hello Opensource.com from  
Java!");  
  
    }  
  
}
```

Hello world in Python

```
print("Hello Opensource.com from Java!")
```

Run the programs

[python-java-hello-world.png](#)

To run the java program "Hello.java" you need to compile it first which creates a "Hello.class" file. To run just the class name, use "java Hello." For Python, you would just run the file "python3 helloworld.py."

6.9 What is Machine Learning?

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed

The primary aim of ML is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

6.9.1 Machine Learning vs. Traditional Programming

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

Machine learning is supposed to overcome this issue. The machine learns how the input and output data are correlated and it writes a rule. The programmers do not need to write new rules each time there is new data. The algorithms adapt in response to new data and experiences to improve efficacy over time.

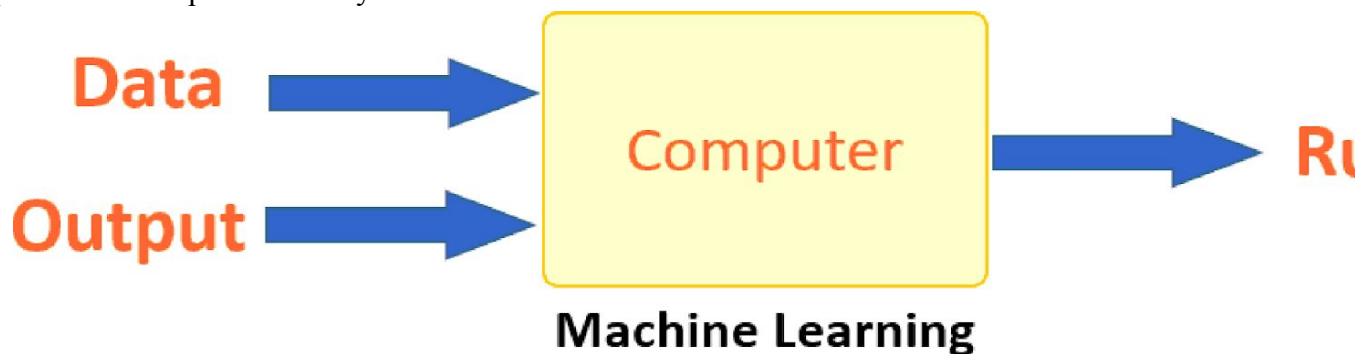


figure 1: Machine Learning.

6.10 How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector**. You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.

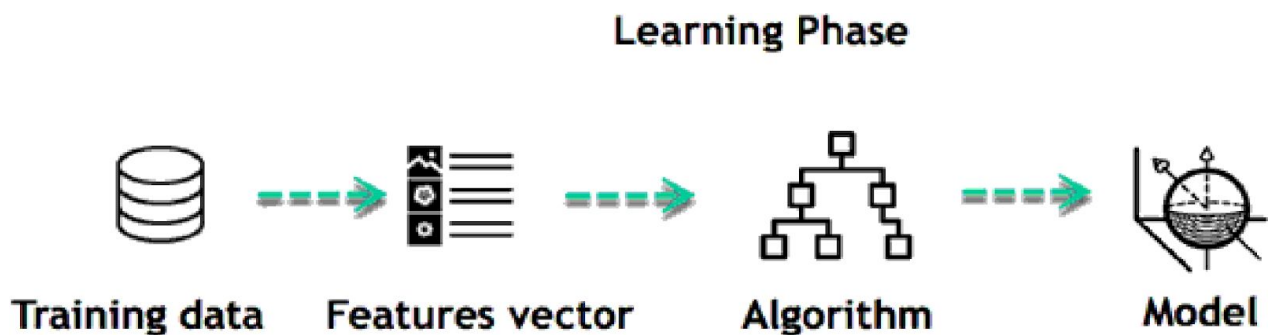


figure 2: Learning Phase

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

The life of Machine Learning programs is straightforward and can be summarized in the following points:

- Define a question
- Collect data
- Visualize data
- Train algorithm
- Test the Algorithm
- Collect feedback
- Refine the algorithm
- Loop 4-7 until the results are satisfying
- Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

6.11 Machine learning Algorithms and where they are used?

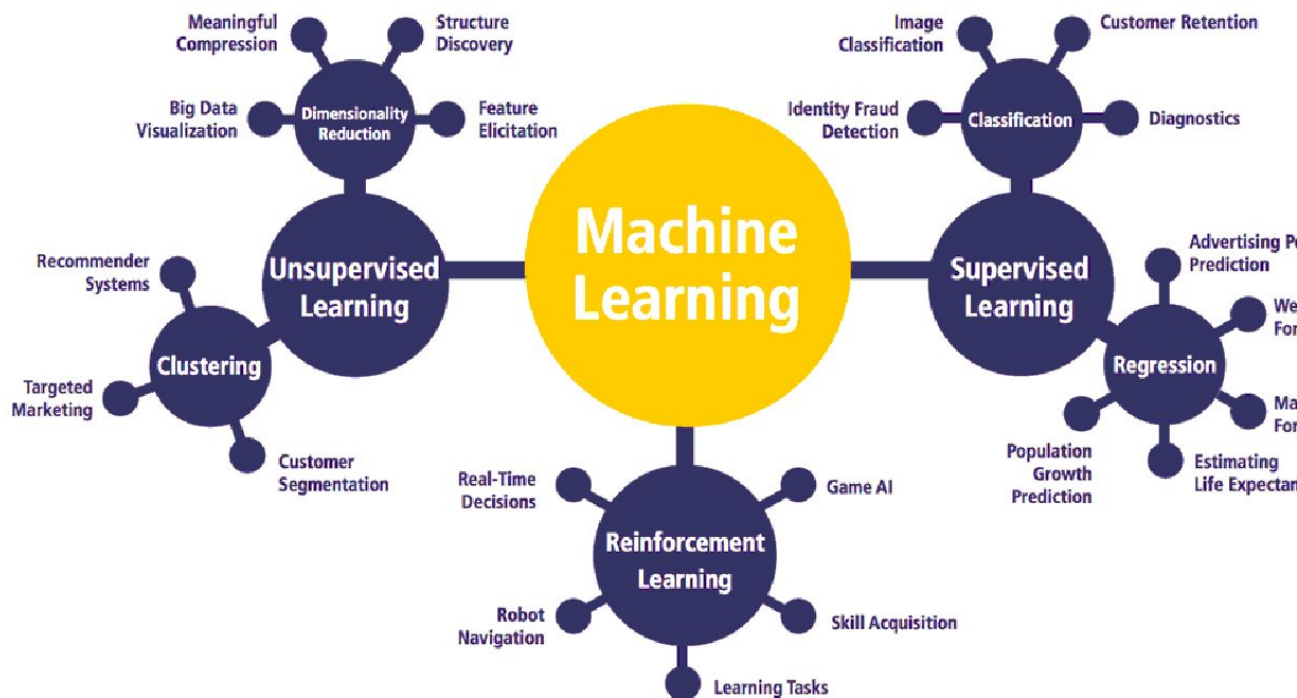


figure 3:Machine Learning Implementation

6.12 Machine Learning is majorly classified in to 3 types

- Supervised Learning
- Unsupervised Learning

Reinforcement learning

6.12.1 Supervised Learning

Supervised learning as the name indicates the presence of a supervisor as a teacher

Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer.

After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data.

Supervised Learning instance

For instance, suppose you are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all different fruits one by one
If shape of object is rounded and depression at top having color Red then it will be labelled as –Apple

If shape of object is long curving cylinder having color Green-Yellow then it will be labelled as –Banana

Now suppose after training the data, you have given a new separate fruit say Banana from basket and asked to identify it

Since the machine has already learned the things from previous data and this time must use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in Banana category

Thus the machine learns the things from training data(basket containing fruits) and then apply the knowledge to test data(new fruit).

Types of supervised learning problems

Supervised learning classified into two categories of algorithms:

Classification: A classification problem is when the output variable is a category, such as “Red” or “blue” or “disease” and “no disease”.

Regression: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

Unsupervised learning

Unsupervised learning is the training of machine using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance

Here the task of machine is to group unsorted information according to similarities, patterns and differences without any prior training of data

Unlike supervised learning, no teacher is provided, that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabeled data by our-self

For instance, suppose it is given an image having both dogs and cats which have not seen ever

Thus the machine has no idea about the features of dogs and cat so we can't categorize it in dogs and cats

But it can categorize them according to their similarities, patterns, and differences i.e., we can easily categorize the above picture into two parts

First part may contain all pics having dogs in it and second part may contain all pics having cats in it. Here you didn't learn anything before, means no training data or examples.

6.12.2 Types of Unsupervised learning problems

Unsupervised learning classified into two categories of algorithms:

Clustering: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior

Association: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y

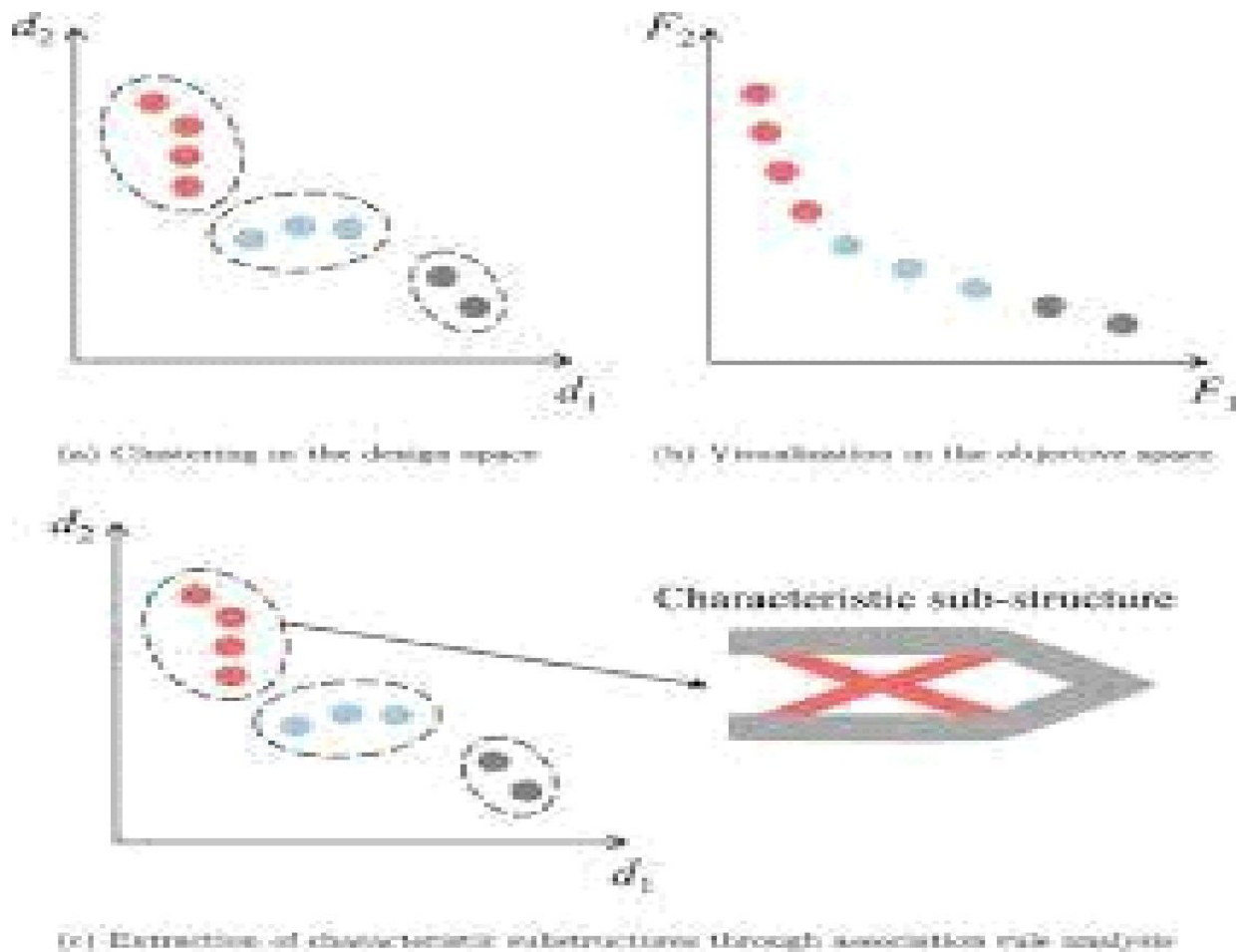


figure 4: Sub Structure

6.12.3 Reinforcement learning

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a situation

It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation

Typical RL scenario

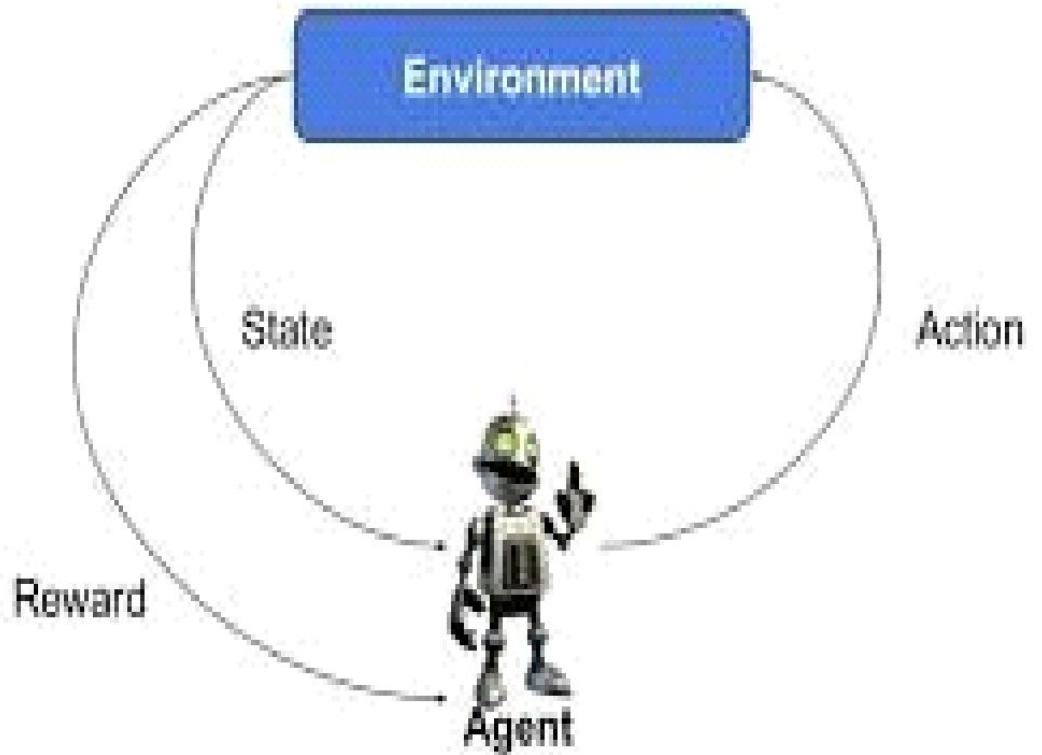


figure 6: Typical RL Scenario

Reinforcement learning with example

We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.

The goal of the robot is to get the reward that is the diamond and avoid the hurdles that is fire (Shown in the figure in the next slide)

The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles

Each right step will give the robot a reward and each wrong step will subtract the reward of the robot

The total reward will be calculated when it reaches the final reward that is the diamond.

Reinforcement learning with example

Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

- Classification task
- Regression task

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

| Algo rithm Name | Description | Type |
|--|---|--|
| Line ar regression | Finds a way to correlate each feature to the output to help predict future values. | Regression |
| Logi stic regression | Extension of linear regression that's used for classification tasks. The output variable is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors) | Classification |
| De cision tree | Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made | Regression Classification |
| Naive Bayes | The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. | Regression Classification |
| Su pport vector machine | Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver. | Regression (not very common) Classification |
| Rando m | The algorithm is built upon a decision tree to improve the | Regression |

| | | |
|--|--|------------------------------|
| forest | accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction. | Classification |
| AdaB oost | Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome | Regression Classification |
| Gra dient- boosting trees | Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it. | Regression Classification |

Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

| Algorithm | Description | Type |
|--------------------------------|--|---------------------|
| K-means clustering | Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans) | Clustering |
| Gaussian mixture model | A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters) | Clustering |
| Hierarchical clustering | Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer | Clustering |
| Recommender system | Help to define the relevant data for making a recommendation. | Clustering |
| PCA/T-SNE | Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances. | Dimension Reduction |

6.13 How to choose Machine Learning Algorithm

There are plenty of machine learning algorithms. The choice of the algorithm is based on the objective.

In the example below, the task is to predict the type of flower among the three varieties. The predictions are based on the length and the width of the petal. The picture depicts the results of ten different algorithms. The picture on the top left is the dataset. The data is classified into three categories: red, light blue and dark blue. There are some groupings. For instance, from the second image, everything in the upper left belongs to the red category, in the middle part, there is a mixture of uncertainty and light blue while the bottom corresponds to the dark category. The other images show different algorithms and how they try to classified the data.

6.14 Challenges and Limitations of Machine learning

The primary challenge of machine learning is the lack of data or the diversity in the dataset. A machine cannot learn if there is no data available. Besides, a dataset with a lack of diversity gives the machine a hard time. A machine needs to have heterogeneity to learn meaningful insight. It is rare that an algorithm can extract information when there are no or few variations. It is recommended to have at least 20 observations per group to help the machine learn. This constraint leads to poor evaluation and prediction.

6.15 Application of Machine learning Augmentation:

Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

Automation:

Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

Finance Industry

Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

Government organization

The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

Healthcare industry

Healthcare was one of the first industry to use machine learning with image detection.

Marketing

Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to

estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

Example of Machine Learning Google Car

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Machine Learning Engineer Salaries in the United States

Salary estimated from 1,801 employees, users, and past and present job advertisements on Indeed in the past 36 months. Last updated: August 24, 2018

Location

United States

Average salary

\$139,026 per year



figure 7 : Average Salaries

6.16 Why is Machine Learning important?

Machine learning is the best tool so far to analyze, understand and identify a pattern in the data. One of the main ideas behind machine learning is that the computer can be trained to automate tasks that would be exhaustive or impossible for a human being. The clear breach from the traditional analysis is that machine learning can take decisions with minimal human intervention.

Take the following example; a retail agent can estimate the price of a house based on his own experience and his knowledge of the market.

A machine can be trained to translate the knowledge of an expert into features. The features are all the characteristics of a house, neighborhood, economic environment, etc. that make the price difference. For the expert, it took him probably some years to master the art of estimate the price of a house. His expertise is getting better and better after each sale.

For the machine, it takes millions of data, (i.e., example) to master this art. At the very beginning of its learning, the machine makes a mistake, somehow like the junior salesman. Once the machine sees all the example, it got enough knowledge to make its estimation. At the same time, with incredible accuracy. The machine is also able to adjust its mistake accordingly.

Most of the big company have understood the value of machine learning and holding data. McKinsey have estimated that the value of analytics ranges from \$9.5 trillion to \$15.4 trillion while \$5 to 7 trillion can be attributed to the most advanced AI techniques.

CHAPTER-7

SAMPLE CODE

7.SAMPLE CODE

```
import numpy as np

import cv2

#import cascades

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

cap = cv2.VideoCapture(0)

fourcc = cv2.VideoWriter_fourcc(*'XVID') #codec

out = cv2.VideoWriter('input.avi',fourcc, 20.0, (640,480))

while 1:

    ret, img = cap.read()

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    #multiScale is used to detect different sizes of images in input.

    faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:

        #w and h represent width and height respectively

        #marking rectangles around the face

        cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

        #Region of Image(ROI) of the face

        roi_gray = gray[y:y+h, x:x+w]
```

```

roi_color = img[y:y+h, x:x+w]

# Using the eye_Cascades to detect the eyes
eyes = eye_cascade.detectMultiScale(roi_gray)
for (ex,ey,ew,eh) in eyes:
    cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
out.write(img)

cv2.imshow('img',img)
k = cv2.waitKey(30) & 0xff
if k == 27:
    img_name = "output.png".format(0)
    cv2.imwrite(img_name, img) #saves the output to the directory
    break

cap.release()
out.release()
cv2.destroyAllWindows()

```

CHAPTER-8

SYSTEM TESTING

8.SYSTEM TESTING

It is a process of establishing confidence that a program or system does what it is proposed of. 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.

8.1 Testing strategy

Unit Testing:

This testing method considers a module as 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 inputs and generate output. Outputs for a given set of input combination are pre-calculated and are generated by the module.

8.2 Integration 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 synchronous with each other. This testing methodology helps in making sure that all modules which are running perfectly when checked individually and are also running cohesion with other modules. For this testing we create test cases to check all modules once and then generated test combinations of test paths through out the system to make sure that no path is making its way into chaos.

8.3 Validation 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 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 endless 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.

8.3 High-order testing (a.k.a. System Testing)

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 in a Depth First manner.

This process is done in Five steps :

Main control module is used as a test driver and steps are submitted are 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 that new errors have not been introduced.

This process continues 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 exist early recognition's 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 upwards in the program structure.

8.4 BOTTOM-UP INTEGRATION TESTING

Begins construction and testing with automatic 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 implement 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 drover's lesions. If the top-levels of the program are integrated top-down, the number of drivers can be reduced substantially and integration of clusters is greatly simplified.

8.5 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 may be conducted manually by executing a subset of all test cases and results for subsequent playback 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.

CHAPTER-9

SCREENSHOTS

9.SCREENSHOTS

9.1 Anaconda Navigator

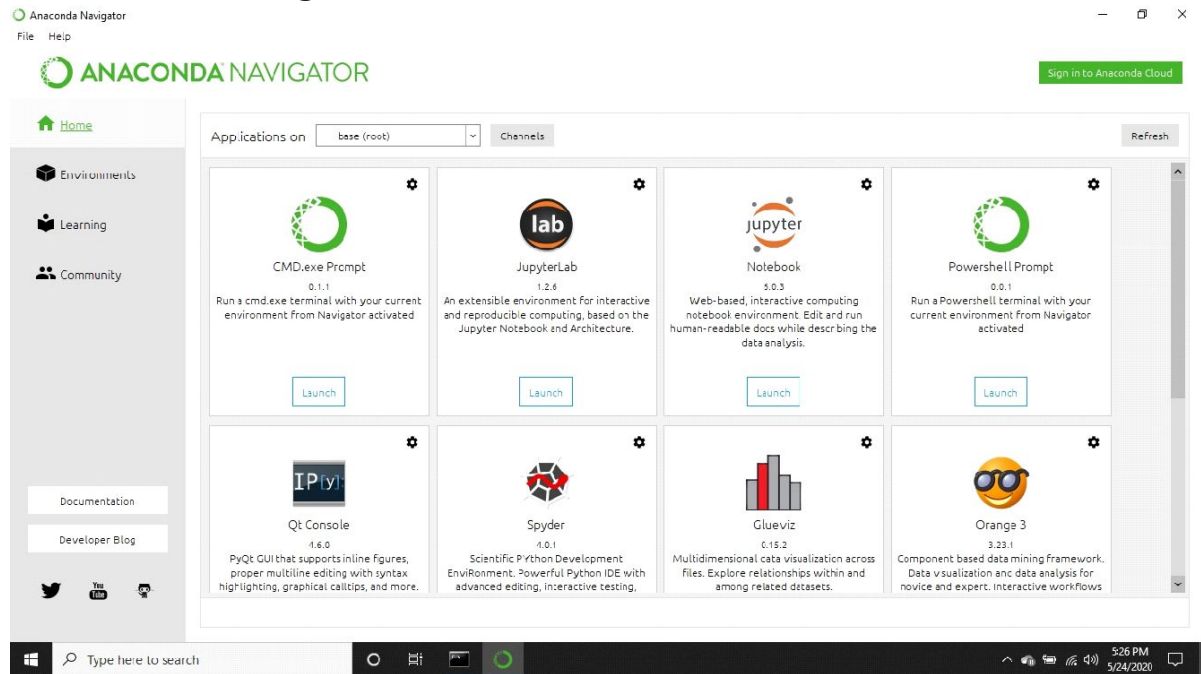
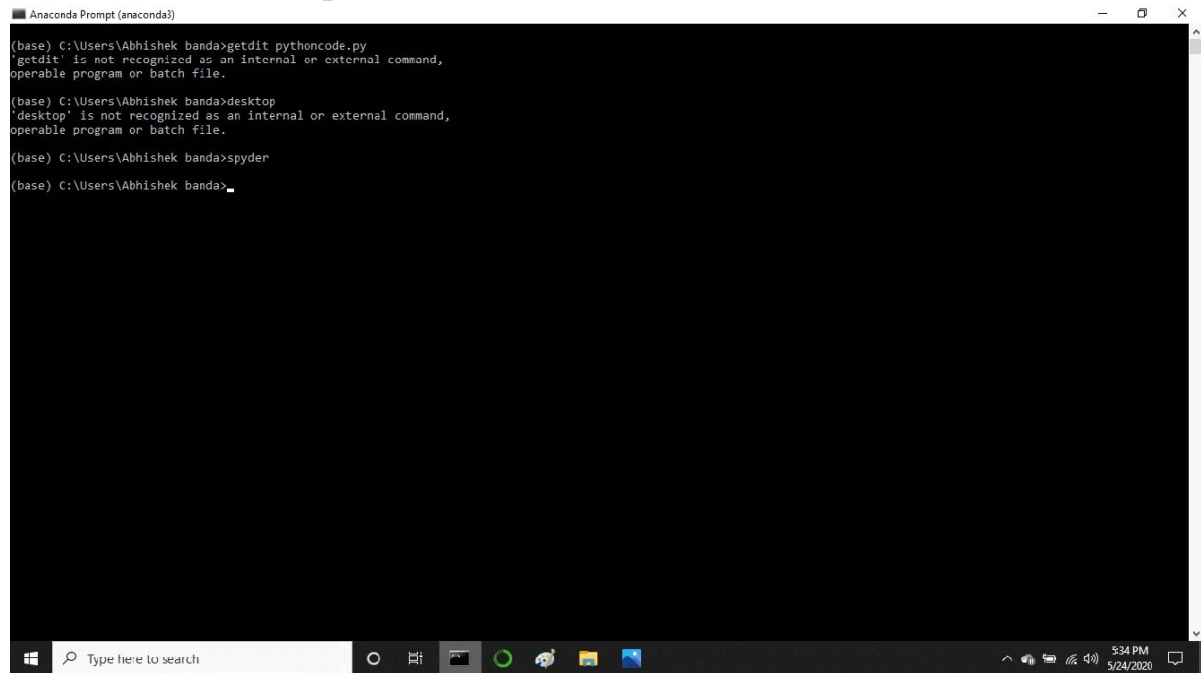


figure 7: Anaconda Navigator

9.2 Anaconda Prompt



The screenshot shows a Windows terminal window titled "Anaconda Prompt (anaconda3)". The prompt is at the root directory "C:\Users\Abhishek banda". The user has entered three commands: "getdit pythoncode.py", "desktop", and "spyder". Each command results in an error message: "'getdit' is not recognized as an internal or external command, operable program or batch file." and "'desktop' is not recognized as an internal or external command, operable program or batch file." The window has a standard Windows taskbar at the bottom with the search bar and system tray.

```
(base) C:\Users\Abhishek banda>getdit pythoncode.py
'getdit' is not recognized as an internal or external command,
operable program or batch file.

(base) C:\Users\Abhishek banda>desktop
'desktop' is not recognized as an internal or external command,
operable program or batch file.

(base) C:\Users\Abhishek banda>spyder
(base) C:\Users\Abhishek banda>
```

figure 8: Anaconda Prompt

9.3 Turning on spyder

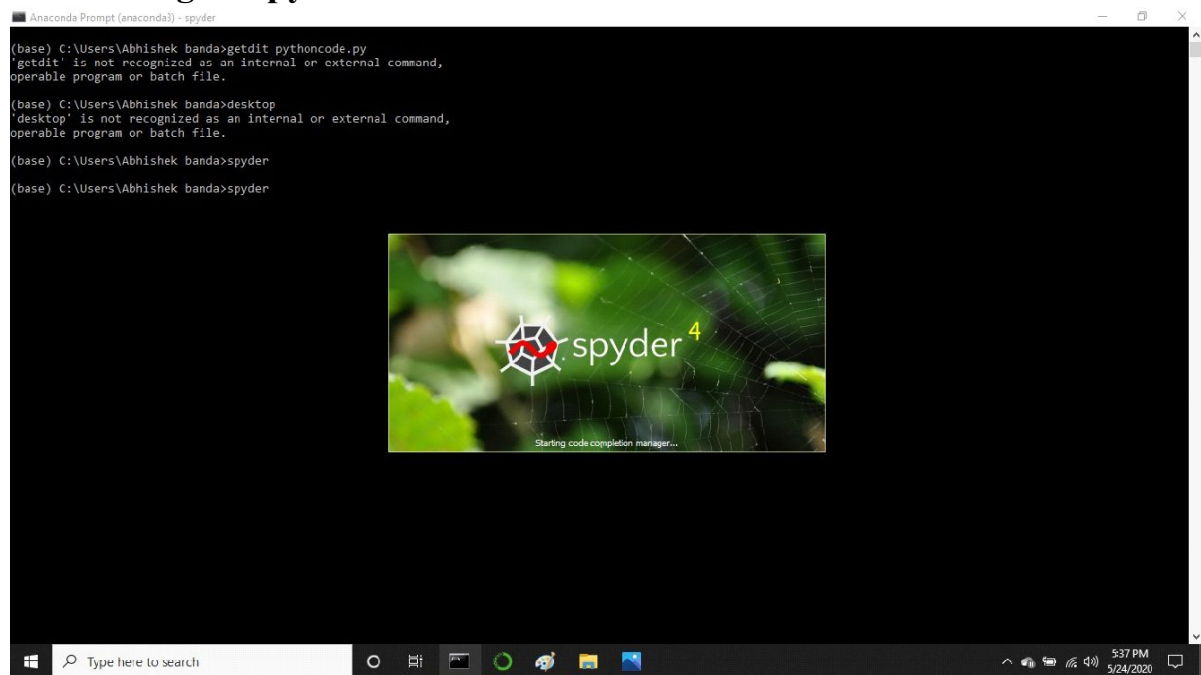


figure 9: Turning On Spyder

9.4 Running Code In Spyder

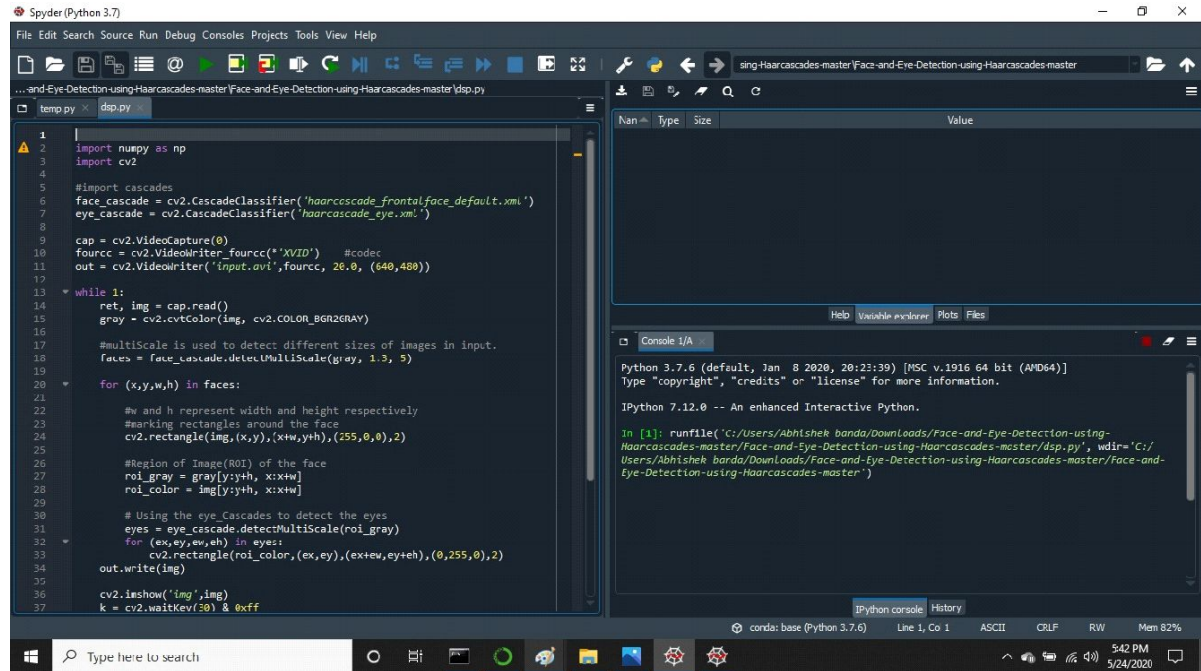


figure 10: Running Code In Spyder

9.5 Files In Spyder

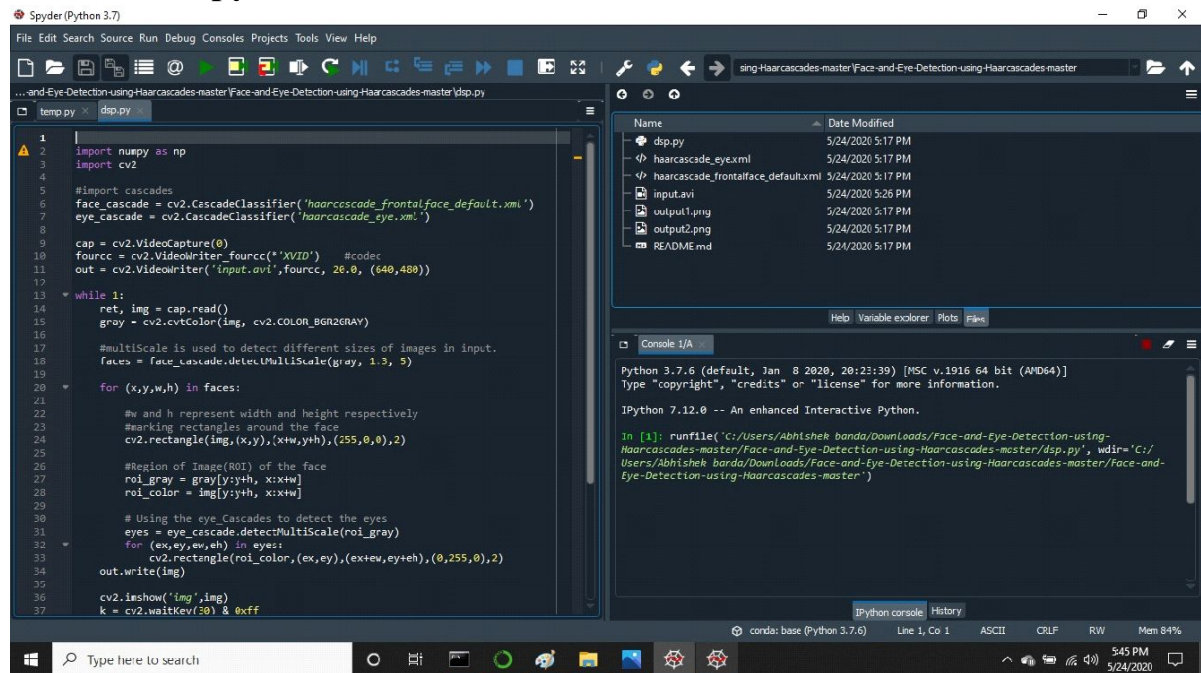


figure 11: Files In Spyder

9.6 Face And Eyes Found

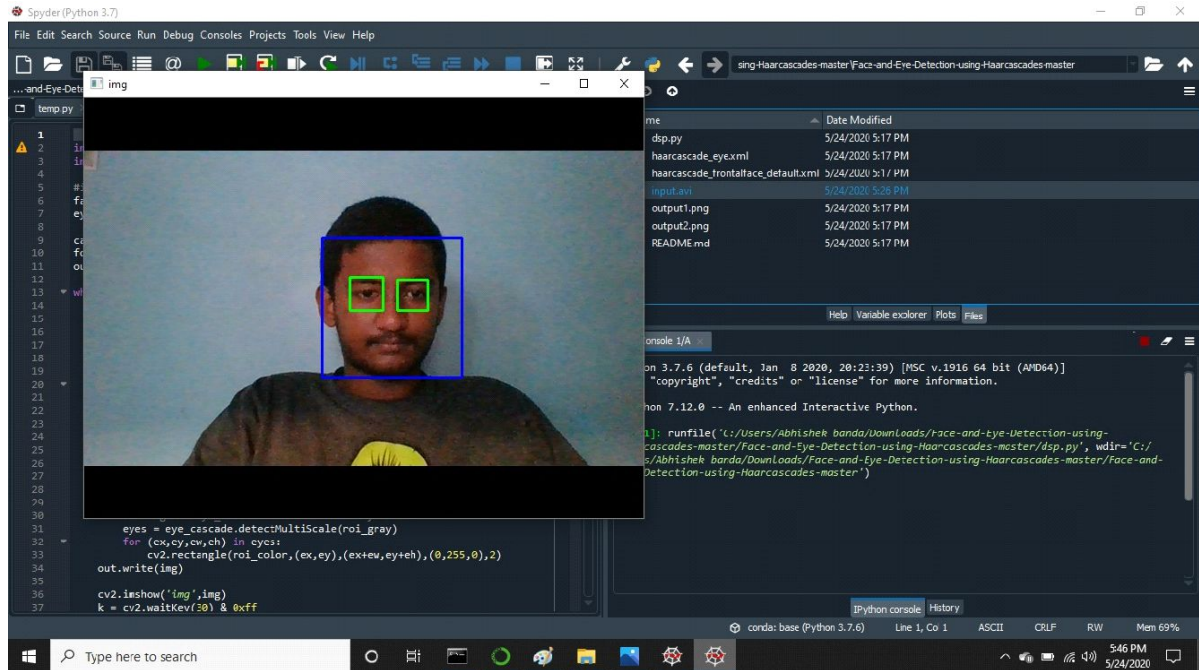


figure 12: Face And Eyes Found

CHAPTER-10

CONCLUSION

10. CONCLUSION

The project has been successfully completed. The goal of the system is achieved and the problems are solved. This project is developed in this manner that is user friendly and required help is provided at different levels the future requirements.

CHAPTER-11

FUTURE ENHANCEMENT

11.FUTURE ENHANCEMENT

In wake of the new and related trends, it is imperative for frequent upgrades to a new models and algorithms to make it easier according to address new business needs.

CHAPTER-12

BIBILOGRAPHY

12.BIBILOGRAPHY

12.1 References

- Paul K. Asabere and Forrest E. Huffman. “Price Concessions, Time of the Market, and the Actual Sale Price of Homes”. In: *Journal of Real Estate Finance and Economics* 6 (1993), pp. 167–174. URL: <https://link.springer.com/article/10.1007/BF01097024>.
- Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.
- Rochard J. Cebula. “The Hedonic Pricing Model Applied to the Housing Market of the City of Savannah and Its Savannah Historic Landmark District”. In: *The Review of Regional Studies* 39.1 (2009), pp. 9–22. URL: journal.srsa.org/ojs/index.php/rss/article/download/182/137.
- Consumer Housing Trends Report 2016. Zillow Group. Accessed: 11/10/2017. 2016. URL: <https://www.zillow.com/research/zillow-group-report-2016-13279/>.
- Harris Drucker et al. “Support vector regression machines”. In: *Advances in neural information processing systems*. 1997, pp. 155–161.
- Gang-Zhi Fan, Seow Eng Ong, and Hian Chye Koh. “Determinants of House Price: A Decision Tree Approach”. In: *Urban Studies* 43.12 (2006), pp. 2301–2315. URL: journals.sagepub.com/doi/pdf/10.1080/00420980600990928.
- Tin Kam Ho. “Random decision forests”. In: *Document analysis and recognition, 1995., proceedings of the third international conference on*. Vol. 1. IEEE. 1995, pp. 278–282.
- Daniel R. Hollas, Ronald C. Rutherford, and Thomas A. Thomson. “Zillow’s estimates of single-family housing values.” In: *Expert Systems with Applications* 78.1 (2010). URL: <http://www.freepatentsonline.com/article/Appraisal-Journal/220765044.html>.
- Gu Jirong, Zhu Mingcang, and Jiang Liuguangyan. “Housing price based on genetic algorithm and support vector machine”. In: *Expert Systems with Applications* 38 (2011), pp. 3383–3386. URL: <http://www.sciencedirect.com/science/article/pii/S0957417410009310>.

- Kelvin J. Lancaster. “A New Approach to Consumer Theory”. In: The Journal of Political Economy 74.2 (1966), pp. 132–157. ISSN: 0303-2647. DOI: 10.1.1.456.4367&rep=rep1&type=pdf. URL: <http://www.jstor.org/stable/1828835>.
- Number of houses sold in the United States from 1995 to 2016. www.statista.com. Accessed: 11/10/2017. URL: <https://www.statista.com/statistics/219963/number-of-us-house-sales/>.
- Quick Facts: Resident Demographics. National Multifamily Housing Council. Accessed: 11/11/2017. 2017. URL: <http://www.nmhc.org/Content.aspx?id=4708>.
- Random Forests by Leo Breiman and Adele Cutler. URL: <https://www.stat.berkeley.edu/~breiman/RandomForests/>.
- Sherwin Rosen. “Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition”. In: The Journal of Political Economy 82.1 (1974), pp. 34–55. URL: <http://people.tamu.edu/HYPERLINKpublicecon/rosen74.pdf>
HYPERLINK "<http://people.tamu.edu/~ganli/publicecon/rosen74.pdf>" ganli/publicecon/rosen74.pdf
HYPERLINK "<http://people.tamu.edu/~ganli/publicecon/rosen74.pdf>" HYPERLINK "<http://people.tamu.edu/~ganli/publicecon/rosen74.pdf>"
- Hasan Selim. “Determinants of house prices in Turkey: Hedonic regression versus artificial neural network”. In: Expert Systems with Applications 36 (2009), pp. 2843–2852. URL: [HYPERLINK "http://www.sciencedirect.com/science/article/pii/S0957417408000596"](http://www.sciencedirect.com/science/article/pii/S0957417408000596)
[HYPERLINK "http://www.sciencedirect.com/science/article/pii/S0957417408000596"](http://www.sciencedirect.com/science/article/pii/S0957417408000596)
- G. Stacy Sirmans, David A. Macpherson, and Emily N. Zietz. “The Composition of Hedonic Pricing Models”. In: Journal of Real Estate Literature 13.1 (2005), pp.3–43. URL: http://www.jstor.org/stable/44103506?seq=1#page_scan_tab_contents.
- Alex J Smola and Bernhard Schölkopf. “A tutorial on support vector regression”. In: Statistics and computing 14.3 (2004), pp. 199–222.
- Danny P. H. Tay and David K. H. Ho. “Artificial Intelligence and the Mass Appraisal of Residential Apartments”. In: Journal of Property Valuation and Investment 10.2 (1992), pp. 525–540. URL: <http://www.emeraldinsight.com/doi/abs/10.1108/14635789210031181>.
- The Price of Overpricing: How Listing Price Impacts Time on Market. Zillow. Accessed: 03/06/2018. 2016. URL: <https://www.zillow.com/research/overpricingimpacts-time-market-12476/>.
- Trulia Estimate. Trulia. Accessed: 11/11/2017. 2017. URL: https://www.trulia.com/trulia_estimates/.
- Vladimir N. Vapnik and Alexey Ya Chervonenkis. “On a class of algorithms of learning pattern recognition.” In: Automation and Remote Control 25.6 (1964).
- Zestimate. Zillow Group. Accessed: 11/11/2017. 2017. URL: <https://www.zillow.com/zestimate/>.