VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI, KARNATAKA

INTERNSHIP TRAINING REPORT

ON

"ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING"

A report submitted in the partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

In

Information Science & Engineering

Submitted by

SURYANSH KUMAR SRIVASTAVA 1SG20IS100

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

SAPTHAGIRI COLLEGE OF ENGINEERING

Bengaluru-57

2023-24

SRI SRINIVASA EDUCATIONAL AND CHARITABLE TRUST®

SAPTHAGIRI COLLEGE OF ENGINEERING

(Affiliated to Visvesvaraya Technological University, Belagavi and Approved by AICTE, NewDelhi)
(Accredited by NAAC with "A" grade)(NBA Accredited-CSE, ECE, EEE, ISE, ME)
(An ISO9001:2015 & ISO14001:2015 Certified)

Department of Information Science & Engineering

CERTIFICATE

Certified that the Internship work entitled **ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING** carried out by **Suryansh Kumar Srivastava (1SG20IS100)** bonafide student of 8th semester, Department of **Information Science & Engineering**, Sapthagiri College of Engineering, Bengaluru in partial fulfillment of the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University,** Belagavi during the year 2023-24. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. The Internship report has been approved as it satisfies the academic requirements in respect of Internship work prescribed for the said Degree.

Prof. Gayathri R	Dr. H R Ranganatha	Dr. H Ramakrishna
Signature of the Guide	Signature of the HOD	Signature of the Principal

Name of the Examiners **Signature with date**

1.

2.

ABSTRACT

This machine learning training provides a comprehensive overview of the principles and practices of machine learning, a rapidly growing field that has transformed the way we approach problem-solving in various industries. Through a combination of theoretical lectures and hands-on exercises, participants will learn how to build intelligent systems that can analyze, learn, and make predictions based on data. The training covers a wide range of topics, including supervised and unsupervised learning, neural networks, deep learning, feature engineering, model evaluation, hyperparameter tuning, and model deployment. Participants will also learn about the ethical considerations involved in machine learning, including issues like bias, fairness, and privacy. By the end of this training, I have gained a strong foundation in machine learning, as well as practical experience working with popular tools and frameworks such as Python, scikit-learn, and TensorFlow.

ACKNOWLEDGEMENT

Any achievement doesn’t depend solely on the individual efforts but on the guidance, encouragement and co-operation of intellectuals, elders and friends. A number of personalities have helped us. I would like to take this opportunity to thank them all.

I would like to express our heart-felt gratitude to **Dr. H Ramakrishna**, Principal, Sapthagiri College of Engineering, Bengaluru, for his help and inspiration during the tenure of the course.

It is great privilege to extend my deep sense of gratitude to **Dr. H R Ranganatha**, Head of the Department, Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru, who patronized throughout my career, for his constant support and encouragement and for the facilities provided to carry out this work successfully.

I also extend my thanks to **Prof. Ambika S,** Internship Coordinator, Dept. of ISE, S.C.E., Bengaluru for her kind cooperation.

With profound sense of gratitude, I acknowledge the guidance and support extended by **Prof. Priyanka M R**, Assistant Professor, Department of Information Science & Engineering, S.C.E., Bengaluru. Her incessant encouragement and valuable technical support have been of immense help in realizing this internship. Her guidance gave us the environment to enhance our knowledge, skills and to reach the pinnacle with sheer determination, dedication, and hard work.

I also extend my sense of gratitude and sincere thanks to all faculty members and non-teaching staff members of Information Science and Engineering, Sapthagiri College of Engineering, Bengaluru for their views and encouraging ideas.

SURYANSH KUMAR SRIVASTAVA (1SG20IS100)

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Introduction About the Company

It is pleasure in introducing "Karunadu Technologies Private Limited" as a leading IT software solutions and services industry focusing on quality standards and customer values. It is also a leading Skills and Talent Development company that is building a manpower pool for global industry requirements.

Fig 1.1: Company Logo

The company offers broad range of customized software applications powered by concrete technology and industry expertise. It also offers end to end embedded solutions and services. They deal with broad range of product development along with customized features ensuring at most customer satisfaction and also empower individual with knowledge, skills and competencies that assist them to escalate as integrated individuals with a sense of commitment and dedication.

1.1.1 Vision

To Empower Unskilled Individual with knowledge, skills and technical competencies in the field of Information Technology and Embedded engineering which assist them to escalate as integrated individuals contributing to company's and Nation's growth.

1.1.2 Mission

- Provide cost effective and reliable solutions to customers across various latest technologies.
- Offer scalable end-to-end application development and management solutions .
- Provide cost effective highly scalable products for varied verticals.
- Focus on creating sustainable value growth through innovative solutions and unique partnerships.
- Create, design and deliver business solutions with high value and innovation by leveraging technology expertise and innovative business models to address long-term business objectives.
- Keep our products and services updated with the latest innovations in the respective requirement and technology.

1.1.3 Values

- To develop software and Embedded solutions and services focusing on quality standards and customer values.
- Offer end to end embedded solutions which ensure the best customer satisfaction.
- To build Skilled and Talented manpower pool for global industry requirements.
- To develop software and embedded products which are globally recognized.
- To become a global leader in Offering Scalable and cost-effective Software solutions and services across various domains like E-commerce, Banking, Finance, Healthcare and more.
- To generate employment for skilled and highly talented youth of our Country INDIA.

1.1.4 Quality Policy

To win consumers confidence and loyalty, a company needs to consistently deliver branded products and items of excellent quality. Karunadu Technologies understand the different needs of the consumers and customers and strive to develop and deliver superior brands to ensure that they are the preferred choice. And by applying consistently high standards, they are able to do things right first time, cut waste, reduce costs and drive profitability.

They have strict mandated quality standards in place, and regular audits. Self- evaluations are used to assess compliance. By adhering to these standards, their business produces items that are great in quality, safe, and compliant with all applicable industry and governmental regulations in the places where they operate. They actively engage with their consumers and customers, translating and incorporating their needs and requirements into their products and services.

1.1.5 Development Sectors

Karunadu Technologies is now establishing connections with academic institutions to provide students with the internships and skill-development programs they need to be prepared for the workforce. They have collaborated with educational institutions in and near Karnataka. The main services offered in the Edtech domains include internships, workshops, pre-placement training, industry-focused lab sets, and hands-on skill development using the most recent technology.

They offer services in a variety of industries, such as healthcare, medical equipment, data analysis and visualization, education, vehicle and driving safety solutions, consumer insights, and smart home appliances. They also extend their services in the areas that include creating mobile applications, developing and hosting websites, using automated transcription, and etc.

1.2 Company Products and Services Offered

1.2.1 Products

- **KECMS – Karunadu Enterprise Content Management System**

Karunadu Enterprise Content Management System is a one stop solution for all our enterprise content management System relating to digital asset management, document imaging, workflow systems and records management systems. Increasing digitalization has led to an exponential growth in business content and managing this sea of unstructured data is tedious work.

- **KEMS – Karunadu Education Management System**

Manage diversified data relating to education management on cloud. Educational data including students and staff is gathered over years which contain information from admission/appointment until leaving the Education. Statistical reports for the College/school can be generated along with admission Tracking and result analysis to keep track of progressive improvements of both student and staff.

- **KASS – Karunadu Advanced Security System**

A Complete one stop embedded solution for large apartments. Security system which monitors door breakage, window breakage, gas leakage, motion detection and various other features which can be operated and maintained by centralized monitored system. This Embedded solution enhances the security measures of apartment/building and enhances the security of individuals may be from unintended intervention or from unauthorized access.

1.3 Overview of the Organization

"Karunadu Technologies Private Limited" is a leading IT software solutions and services industry focusing on quality standards and customer values. It is also a leading Skills and Talent Development company that is building a manpower pool for global industry requirements.

The company offers broad range of customized software applications powered by concrete technology and industry expertise. It also offers end to end embedded solutions and services. They deal with broad range of product development along with customized features ensuring at most customer satisfaction and also empower individual with knowledge, skills and competencies that assist them to escalate as integrated individuals with a sense of commitment and dedication.

1.3.1 Services

- **IT Solutions and Services**

Karunadu Technologies is a Bangalore based IT Training and Software Development center with an exclusive expertise in the area of IT Services and Solutions. Karunadu Technologies Pvt. Ltd. is also expertise in Web Designing and Consulting Services.

- **Embedded Design and Development**

Karunadu Technologies Pvt. Ltd. has expertise in Design and development of embedded products and offers solutions and services in field of Electronics.

- **Academic Projects**

Karunadu Technologies Pvt. Ltd. helps students in their academics by imparting industrial experience into projects to strive excellence of students. Karunadu Technologies Pvt. Ltd. encourages students to implement their own ideas to projects keeping in mind "A small seed sown upfront will be nourished to become a large tree one day", thereby focusing the future entrepreneurs. They have a wide range of IEEE projects for B.E, MTech, MCA, BCA, DIPLOMA students for all branches in each and every domain.

- **Embedded Design and Development**

Karunadu Technologies Pvt. Ltd. has expertise in Design and development of embedded products and offers solutions and services in field of Electronics.

- **Academic Projects**

Karunadu Technologies Pvt. Ltd. helps students in their academics by imparting industrial experience into projects to strive excellence of students. Karunadu Technologies Pvt. Ltd. encourages students to implement their own ideas to projects keeping in mind "A small seed sown upfront will be nourished to become a large tree one day”, thereby focusing the future entrepreneurs. They have a wide range of IEEE projects for B.E, MTech, MCA, BCA, DIPLOMA students for all branches in each and every domain.

- **Implant Training**

Karunadu Technologies Pvt. Ltd. provides Implant training for students according to the interest of students keeping in mind the current technology and academic benefit one obtains after completing the training. Students will be nourished and will be trained throughout with practical experience. Students will be exposed to industrial standards which boost their carrier. Students will become Acquaint to various structural partitions such as labs, workshops, assembly units, stores, and administrative unit and machinery units. They help students to understand their functions, applications and maintenance. Students will be trained from initial stage that is from collection of Project Requirements, Project Planning, Designing, implementation, testing, deployment and maintenance there by helping to understand the business model of the industry. Entire project life cycle will be demonstrated with hands on experience. Students will also be trained about management skills and team building activities.

- **Software Courses**

Karunadu Technologies Pvt. Ltd. provides courses for students according to the interest of students keeping in mind the current technology and assist them for their further Employment. Company provides various courses such as C, C++, VB, DBMS, Dot Net, Core Java and J2EE along with live project. These courses serve as the foundation for students to develop proficiency in software development, covering fundamental programming principles, database management, web development frameworks, and enterprise-level application architecture. Emphasizing practical learning, the company offers opportunities for students to engage in live projects, enabling them to apply theoretical knowledge to real-world scenarios. Through hands-on experience and expert guidance, students not only acquire technical skills but also gain valuable insights into industry best practices.

CHAPTER 2

TASKS PERFORMED

2.1 Learning Experience

The value that I have gained is to always work hard even if the task is small and it seems unimportant. It helped me to build a good work idea, and the effort could be seen. My co-workers had a lot of experience I have talked to them and asked for some advice they have for me. I could learn a lot and get more ideas. I think this internship is extremely cherished by me. The internship enhanced my skill and ability to work in a team.

It also means that we have learned many things. My strength in the internship is that I am a good team builder. As a member of the team, I am responsible in group discussions and giving my own opinions. If there was anything that I am not able to understand I would ask, besides I am a cooperative person.

2.2 Knowledge Acquired

The knowledge I have gained in our training is about Artificial Intelligence and Machine Learning. I have learnt many things about the science of collecting, analyzing and preprocessing raw data to train my machine learning model effectively. Machine learning models can process vast amounts of data quickly and accurately, making it possible to analyze and understand complex data sets that would be impossible for humans to process on their own. Machine learning models can make accurate predictions and forecasts based on historical data, allowing businesses to make informed decisions and plan for the future.

It has been a great technical learning experience as it is thought to me a lot about various machine learning algorithms and appropriate use of it really improved my technical knowledge and skills.

2.3 Skills Learned

I have learned to work well in a team. Another side that I learned throughout my internship is to never be afraid to ask doubts. By asking questions I got answers. I learnt to read the use cases and understand the need and analyze the problem and find a solution to it. I had worked on projects in groups and group discussion helped me a lot in finding the solution because teamwork helps to get more ideas. Internships give students the hands-on experience they need.

2.3.1 Python

Python is a multiparadigm, general-purpose, interpreted, high-level programming language. Python allows programmers to use different programming styles to create simple or complex programs, get quicker results and write code almost as if speaking in a human language.

2.3.2 NumPy

NumPy is basic package for scientific computing. It is the python language implementation which includes powerful N-dimensional array structure, sophisticated functions, Tools that can be integrated into C/C++ and Fortran code, Linear algebra, Fourier transform and Random number features. Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data.

NumPy's extensive library of mathematical functions provides tools for array manipulation, linear algebra, Fourier analysis, and more, making it indispensable in fields like data science, machine learning, and scientific computing. Its seamless integration with other Python libraries, such as SciPy and Matplotlib, further extends its capabilities, enablingcomprehensive data analysis and visualization workflows. With its speed, versatility, and robustness, NumPy remains a fundamental tool for tackling the computational challenges of modern data-driven research and applications.

- **Basic Array Operations**

NumPy, arrays allow a wide range of operations which can be performed on a particular array or a combination of Arrays. These operations include some basic Mathematical operation as well as Unary and Binary operations.

```
# Python program to demonstrate
# basic operations on single array
import NumPy as np a = np. array ([[1, 2], [3, 4]]) # Defining Array 1
b = np.array([[4, 3], [2, 1]]) # Defining Array 2
print ("Adding 1 to every element:", a + 1) # Adding 1 to every element
print ("\n Subtracting 2 from each element:", b - 2) # Subtracting 2 from each element
# sum of array elements print ("\n Sum of all array "elements: ", a.sum()) # Performing Unary operations
print ("\n Array sum:\n", a + b) # Performing Binary operations
```

2.3.3 Pandas

Pandas Data Frame is two-dimensional size-mutable, potentially heterogeneous tabular data structure with labelled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns. Pandas Data Frame consists of three principal components, the data, rows, and columns.

With Pandas, tasks such as data cleaning, transformation, and exploration become streamlined and intuitive. Its rich set of functions facilitates operations like indexing, grouping,and aggregation, empowering users to extract valuable insights from their datasets with minimal effort. Whether handling small-scale datasets or big data, Pandas remains a versatile tool trusted by data scientists, analysts, and researchers alike for its robustness and flexibility. The basic operations which can be performed on Pandas Data Frame are:

- Creating a Data Frame
- Dealing with Rows and Columns
- Indexing and Selecting Data
- Working with Missing Data

2.3.4 Matplotlib

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar charts, etc.

Matplotlib stands as a cornerstone in the Python data visualization landscape, offering a versatile toolkit for creating a wide array of plots and charts. With its intuitive interface and extensive customization options, Matplotlib empowers users to visually explore their data withease. Whether generating simple line plots, intricate scatter plots, or sophisticated heatmaps, Matplotlib remains a go-to library for data visualization tasks in fields ranging from scientific research to financial analysis. Its seamless integration with other Python libraries, such as NumPy and Pandas, further enhances its utility, making it an indispensable tool in the data scientist's toolkit. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar charts, etc,

```
#Python program using Matplotlib for forming a linear plot
import matplotlib. pyplot as plt          # importing the necessary packages and modules
import NumPy as np
x = np. linspace (0, 10, 100)             # Prepare the data
plt. Plot (x, x, label ='linear' )        # Plot the data
plt. Legend()                             # Add a legend
plt. Show()                               # Show the plot
```

2.3.5 Open CV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java,etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as NumPy which is a highly optimizedlibrary for numerical operations, then the number of weapons increases in your Arsenal i.e., whatever operations one can do in NumPy can be combined with OpenCV.

Definition

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

Here we convert the given image as shown in 2.1 into a grey scale image i.e., Black and white image. The function used here is cvt color and the code used for grey scale is 'COLOR_BGR2GRAY'. Similarly, we do for HSV scale and the function used is 'COLOR_BGR2HSV'. Following is anexample for the above scales.

CODE:

```
Import cv path="C:\\Users\\Lenovo \\OneDrive\\Desktop\\MLINTERN\\opencvimages\\Im
ages\\Image1.png"
img = cv2.imread(path)
cv2.imshow("Original Image", img)
img1= cv2.resize(img,(512,512))
cv2.imshow("Resized Image",img1)
graying = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
```

```
graying= cv2.resize(grayimg,(512,512))
cv2.imshow("Gray Image", grayimg)
havimg=cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
havimg=cv2.resize(havimg,(512,512))
cv2.imshow(" HSV Image", havimg)
cv2.waitKey(0) cv2.destroyAllWindows()
```

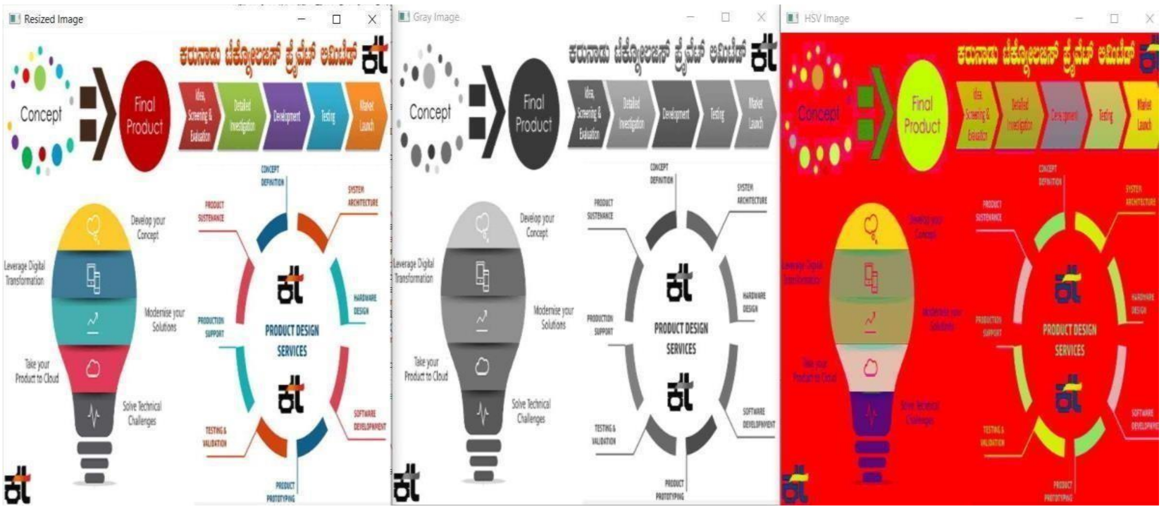


Fig 2.1: Example of Grey and HSV Image

2.3.6 Machine Learning

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict newoutput values. a subset of artificial intelligence, revolutionizes how computers learn and makedecisions from data without explicit programming. At its core, machine learning algorithms extract patterns and insights from vast datasets, enabling systems to improve performance over time through experience. Supervised learning algorithms learn from labeled data to make predictions or classifications, while unsupervised learning algorithms discover hidden patterns and structures within unlabeled data. Reinforcement learning algorithms, inspired by behavioral psychology, enable agents to learn optimal actions through trial and error in dynamic environments. Machine learning finds applications across diverse domains, including image and speech recognition, natural language processing, recommendation systems, and autonomous vehicles. Its rapid advancement, fueled by big data, computational power, and algorithmic innovations, drives transformative changes in industries, shaping the future of technology and society.

The goal of ML, in simple words, is to understand the nature of human and other forms of learning, and to build learning capability in computers. To be more specific, there are three aspects of the goals of ML. To make the computers smarter, more intelligent. The more direct objective in this aspect is to develop systems (programs) for specific practical learning tasks in application domains. To develop computational models of human learning process and perform computer simulations.

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. The Fig 2.2 shows the machine learning workflow: data is collected, processed, used to train a model, deployed, and then monitored for continued accuracy.

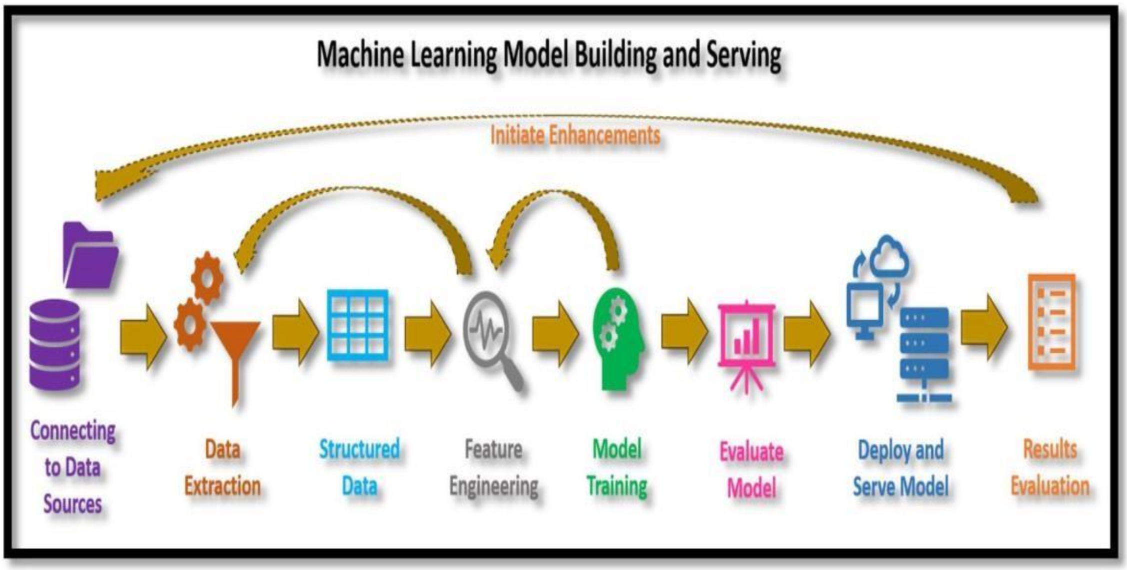


Fig 2.2: Machine Learning Infrastructure

2.3.7 Applications of Machine Learning

Machine learning has been recognized as central to the success of Artificial Intelligence, and it has applications in various areas of science, engineering and society. Someof them are:

- Product recommendations (e.g., Amazon etc.)
- Refining the search engine results (e.g., Google)
- Fighting the web spam (e.g., Gmail)
- Video surveillance (e.g., crime alerts)

- Face recognition
- Predicting future values based on sequences of data points.
- Reducing the number of features in a dataset while preserving essential characteristics.
- Analysing data points in images or frames for tasks like object detection and facial recognition.
- Using patient data points for tasks like disease diagnosis and personalized treatment recommendation.

2.3.8 Types of Machine Learning

- **Supervised learning**

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output. The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**. The Fig 2.3 outlines the two main categories of supervised learning algorithms: regression and classification. Regression algorithms, like linear regression or random forest, excel at predicting continuous values, such as weather patterns or stock prices. Classification algorithms, on the other hand, focus on predicting categorical outputs, like image recognition (identifying objects in a picture) or spam filtering (separating legitimate emails from unwanted messages). Examples of classification algorithms include logistic regression and decision trees. The diagram also acknowledges convolutional neural networks (CNNs) in the lower right corner, which are powerful tools that can be applied to both regression and classification tasks.

The process involves iterative adjustment of model parameters during the training phase to minimize the disparity between predicted and actual labels. Model selection, hyperparameter tuning, and feature engineering are crucial steps in optimizing performance. Techniques such as ensemble methods, transfer learning, and addressing imbalanced data distributions further enhance model robustness and accuracy. However, ethical considerations such as fairness, bias, and privacy must be carefully addressed. Continual learning capabilities are also increasingly important, allowing models to adapt to evolving data distributions over time. Supervised learning finds applications across diverse domains, ranging from image and speech recognition to medical diagnosis and recommendation systems, driving many real- world machine learning advancements.

With the advent of deep learning, neural networks have become a dominant force in supervised learning, achieving state-of-the-art performance in tasks like image recognition, natural language processing, and speech recognition. Supervised learning finds applications in diverse fields such as healthcare, finance, marketing, and robotics, driving innovation and improving decision-making processes.

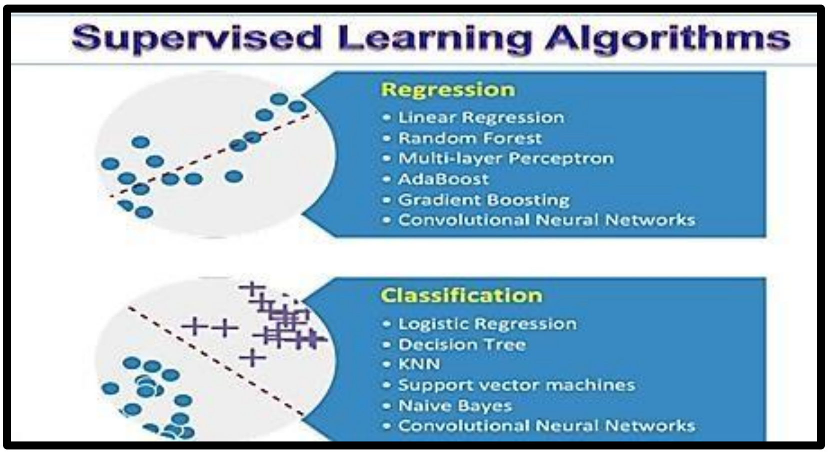


Fig 2.3: Flow Model of Algorithms of Supervised Learning

Supervised learning can be grouped further in 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 labelled 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, machine has no any 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 as shown in Fig 2.4.

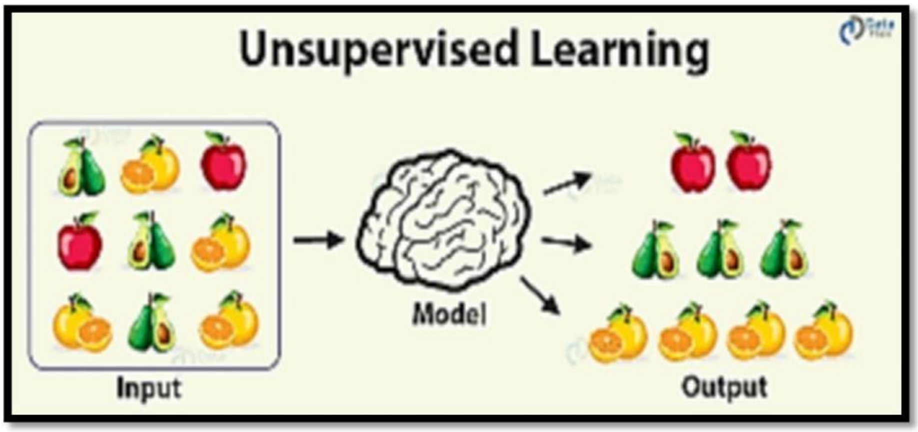


Fig 2.4: Flow Model of Algorithms of Unsupervised Learning

Dimensionality reduction techniques, like principal component analysis (PCA) and t- distributed stochastic neighbor embedding (t-SNE), aim to compress high-dimensional data into lower-dimensional representations while preserving its essential characteristics. Other unsupervised learning methods include anomaly detection, which identifies unusual or unexpected patterns in data, and association rule learning, which discovers relationships between variables in large data sets. Unsupervised learning plays a crucial role in exploratory. data analysis, feature engineering, and preprocessing tasks, aiding in data understanding and providing valuable insights into complex datasets across various domains.

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 restrictedto 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, machine has no any 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.

The Fig 2.5 is a diagram comparing three machine learning tasks: classification, regression, and clustering.

- Classification is the task of learning a function that maps an input to a discrete output. In the image, this is illustrated by classifying an email into spam or not spam (C2). Classification algorithms are used for a variety of tasks, including image recognition, fraud detection, and sentiment analysis.

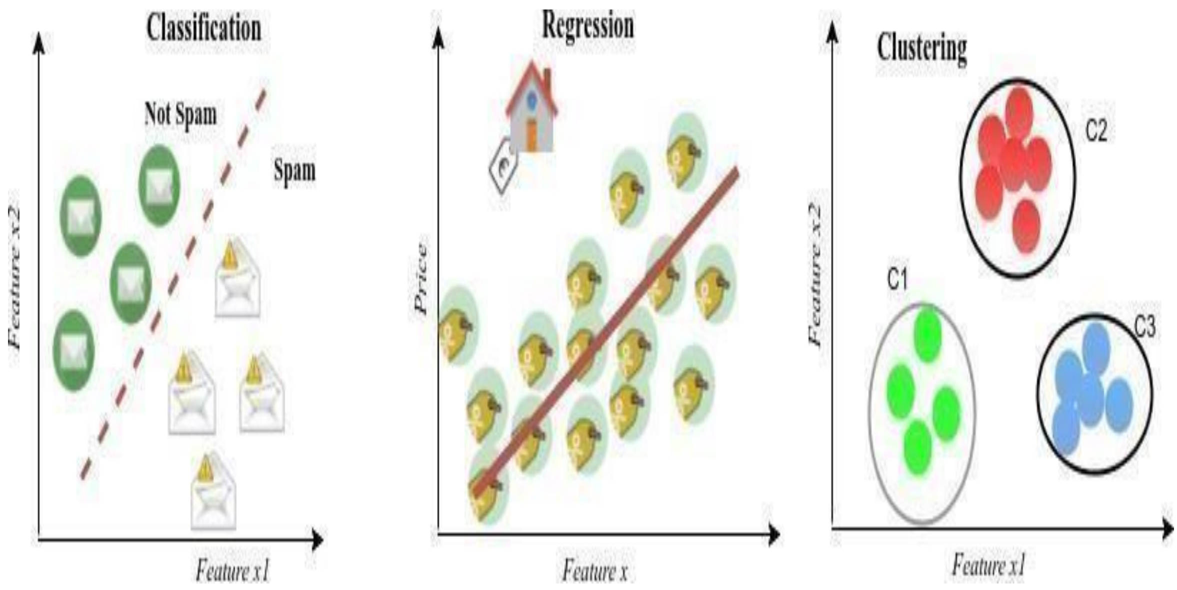


Fig 2.5: Graph of classification, regression and clustering

- Regression is the task of learning a function that maps an input to a continuous output. In the image, this is illustrated by predicting the price of a house based on its size and location (Price). Regression algorithms are used for a variety of tasks, including weather forecasting, stock price prediction, and customer churn prediction.
- Clustering is the task of grouping a set of data points into groups (clusters) such that data points in the same cluster are more similar to each other than data points in different clusters. In the image, this is illustrated by grouping customers into different clusters based on their purchasing habits (C1, C2, C3). Clustering algorithms are used for a variety of tasks, including market segmentation, anomaly detection, and image segmentation.

The key difference between classification and regression is that classification predicts discrete outputs, while regression predicts continuous outputs. Clustering is different from both classification and regression in that it does not predict any outputs. Instead, it groups data points together based on their similarity.

The detailed explanation for Fig: 2.5 given below:

- The x-axis represents two features, which are variables used to describe the data. In the image, the features are not specified but could be things like a customer's age and income.
- The y-axis represents the output. In the case of classification, the y-axis represents the different classes. In the case of regression, the y-axis represents the continuous output. In the case of clustering, there is no y-axis.

- **Reinforcement Learning**

Reinforcement learning is an area of Machine Learning. It is about taking suitable actionto maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas inreinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience.It is a dynamic and powerful paradigm in machine learning that enables agents to learn optimalbehavior through interaction with an environment.

Unlike supervised and unsupervised learning, reinforcement learning does not rely on labeled data but instead learns from feedbackin the form of rewards or penalties received based on actions taken. The agent explores the environment by taking actions and receives feedback in the form of rewards, allowing it to learn which actions lead to desirable outcomes. Through trial and error, reinforcement learningalgorithms aim to maximize the cumulative reward over time, discovering optimal strategiesor policies for various tasks. Reinforcement learning has applications in a wide range of domains, including robotics, game playing, autonomous vehicles, recommendation systems, finance, and healthcare. Its ability to learn from interactions with the environment makes it particularly well-suited for problems where explicit supervision or large labeled datasets.

Key components of reinforcement learning include the agent, which takes actions based on its current state; the environment, which responds to the agent's actions and transitions to new states; and the reward signal, which provides feedback on the agent's performance. Reinforcement learning has applications in a wide range of domains, including robotics, gameplaying, autonomous vehicles, and resource management, where agents learn to make decisions in complex and dynamic environments to achieve long-term goals.

Fig 2.6 depicts flow model of algorithms of Reinforcement learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from the supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of training dataset, it is bound to learn from its experience.

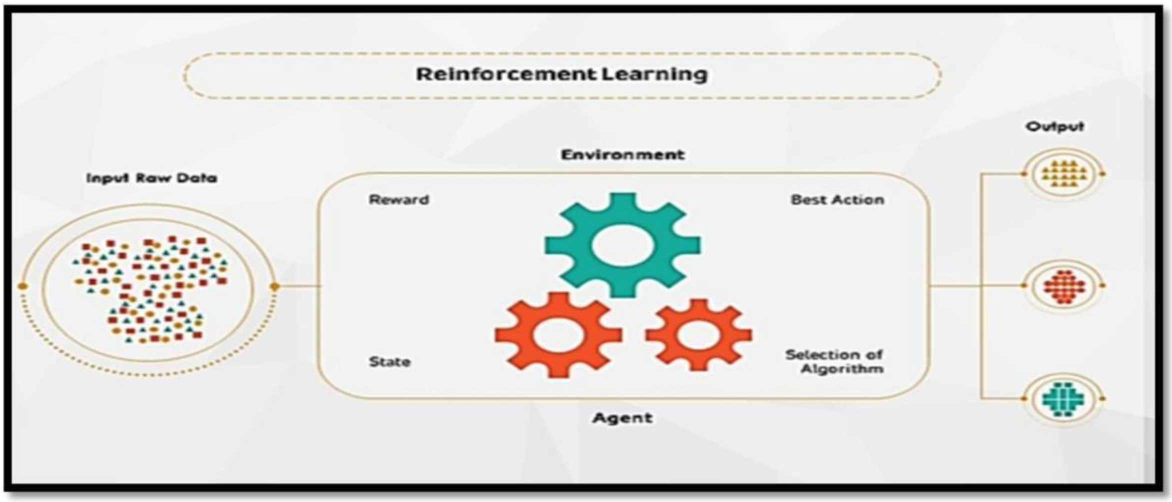


Fig 2.6: Flow Model of Algorithms of Reinforcement Learning

2.3.9 Machine Learning Algorithms

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. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves as shown in Fig 2.7.

These algorithms encompass a diverse range of techniques and approaches, each suited to different types of problems and data. Supervised learning algorithms learn from labeled datasets, where the inputs are paired with corresponding outputs, enabling them to make predictions or classifications on unseen data. Unsupervised learning algorithms, on the other hand, explore unlabeled data to discover hidden patterns or structures, clustering similar data points or reducing the dimensionality of the feature space. Reinforcement learning algorithms learn through trial and error, interacting with an environment to maximize a reward signal.

Additionally, semi-supervised and self-supervised learning algorithms leverage both labeled and unlabeled data or use the data itself to generate supervision signals, respectively. With the advent of deep learning, neural network-based algorithms have gained prominence, revolutionizing fields like computer vision, natural language processing, and speech recognition. Overall, machine learning algorithms play a pivotal role in extracting insights from data, automating tasks, and driving innovation across various domains. Classification is like sorting emails into spam or not spam, while regression is like predicting house prices. Clustering groups similar data points together, such as grouping customers based on buying habits. The algorithms include Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Decision Trees, Random Forests, etc.

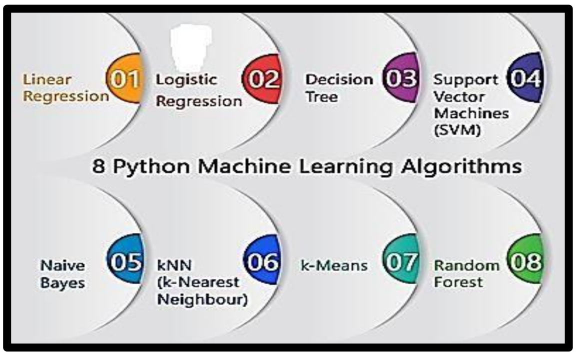


Fig 2.7: Types of Machine Learning Algorithms

- **Linear Regression**

It is one of the most widely known modeling technique. Linear regression is usually among the first few topics which people pick while learning predictive modelling. In this technique, the dependent variable is continuous, independent variable(s) can be continuous or discrete, and nature of regression line is linear. Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables(X) using a best fit straight line (also known as regression line) as shown in Fig 2.8. It is represented by an equation Y=a+b*X + e, where a is intercept, b is slope of the line and e is error term. This equation can be used to predict the value of target variable based on given predictor variable.

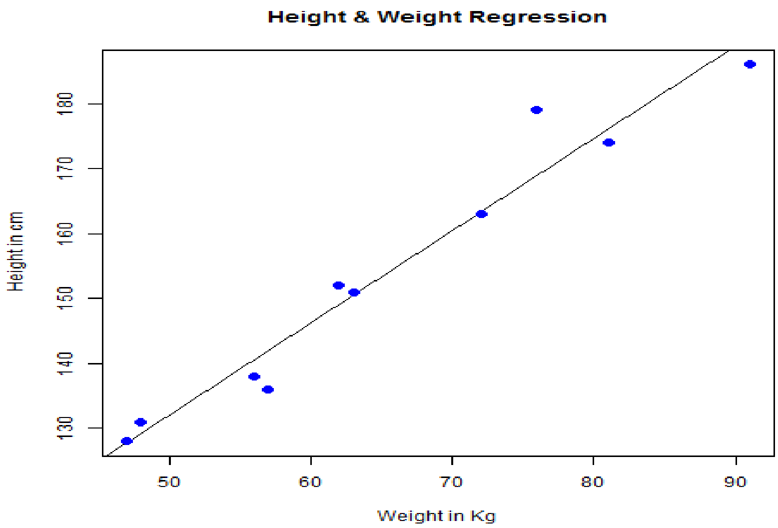


Fig 2.8: Linear Regression

There must be linear relationship between independent and dependent variables. Linear Regression is very sensitive to Outliers. It can terribly affect the regression line and eventually the forecasted values. Simple linear regression is used for finding the relationship between the dependent variable Y and the independent or predictor variable X. Both of these variables are continuous in nature. While performing simple linear regression, we assume thatthe values of predictor variable X are controlled. Furthermore, they are not subject to the measurement error from which the corresponding value of Y is observed.

- **Advantages and Disadvantages**

Advantages

Linear regression is an extremely simple method. It is very easy and intuitive to use and understand. A person with only the knowledge of high school mathematics can understand and use it. In addition, it works in most of the cases. Even when it doesn't fit the data exactly, we can use it to find the nature of the relationship between the two variables.

Disadvantages

• By its definition, linear regression only models relationships between dependent and independent variables that are linear. It assumes there is a straight-line relationship between them which is incorrect sometimes. Linear regression is very sensitive to the anomalies in the data (or outliers).

• Take for example most of your data lies in the range 0-10. If due to any reason only one of the data items comes out of the range, say for example 15, this significantly influences the regression coefficients.

- **Multiple Linear Regression**

In many cases, there may be possibilities of dealing with more than one predictor variable for finding out the value of the response variable. Therefore, the simple linear models cannot be utilized as there is a need for undertaking multiple linear regression for analyzing the predictor variables. The difference between simple linear regression and multiple linear regression is that, multiple linear regression has more than 1 independent variables, whereas simple linear regression has only 1 independent variable. Using the two explanatory variables, we can delineate the equation of multiple linear regression as follows:

$$y_i = b_0 + b_1x_1 + bx_2 + e_i$$

The two explanatory variables $x1$ and $x2$, determine yi, for the ith data point. Furthermore, the predictor variables are also determined by the three parameters $b0$, $b1$, and $b2$ of the model, and by the residual ei of the point i from the fitted surface. General Multiple regression models can be represented as:

$$yi = \sum bi\ xi + ei$$

Multiple regression suffers from multicollinearity, autocorrelation, heteroskedasticity. Multicollinearity can increase the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model. The result is that the coefficient estimates are unstable. In case of multiple independent variables, we can go with forward selection, backward elimination and step wise approach for selection of most significant independent variables.

The coefficients are estimated using methods like ordinary least squares (OLS) to minimize the sum of squared differences between the observed and predicted values. Multiple regression allows for the examination of the unique contribution of each independent variable while controlling for the effects of others, making it a valuable tool in various fields, including economics, social sciences, and business analytics, for modeling complex relationships and making predictions based on multiple factors.

- **Advantages and disadvantages**

Advantages

• The ability to determine the relative influence of one or more predictor variables to the criterion value.

• The ability to identify outliers, or anomalies.

Disadvantages

• Any disadvantage of using a multiple regression model usually comes down to the data being used. Two examples of this are using incomplete data and falsely concluding that a correlation is a causation.

• Adding too many predictor variables can increase the model's complexity and lead to overfitting. This means the model performs well on the training data but struggles to generalize to unseen data.

• Multiple linear regression assumes a linear relationship between the independent variables and the dependent variable. If the relationships are non-linear, the model won't capture the true underlying process.

• While powerful, multiple linear regression can struggle with very large datasets due to computational limitations and memory usage.

• With many predictor variables, it can become difficult to understand the individual contribution of each variable to the model's predictions.

- **Logistic Regression**

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model (a form of binary regression). The Fig 2.9 is a comparison of logistic regression and linear regression. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an indicator variable, where the two values are labelled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labelled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labelled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labelling, the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a logit, from logistic unit, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the profit model; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio.

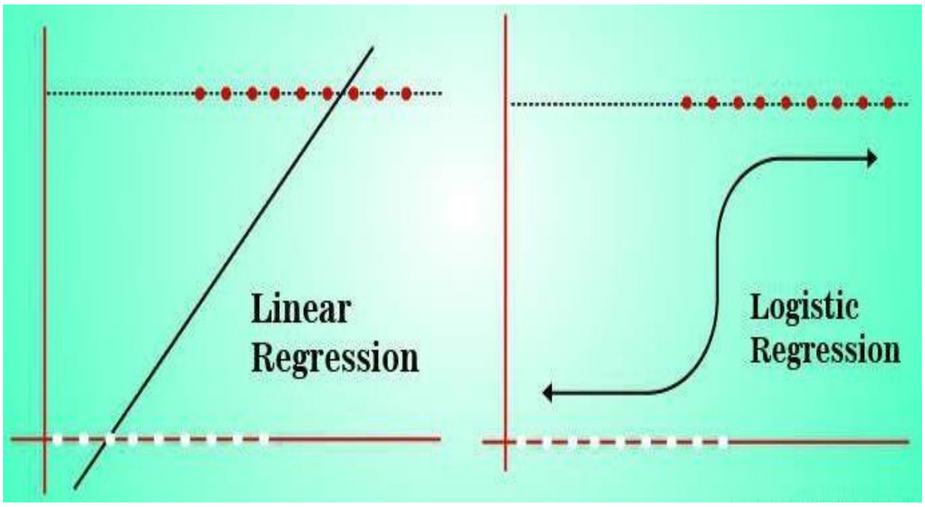


Fig 2.9: Linear Regression v/s Logistic Regression

- **Key Features**

  • Logistic regression predicts whether something is True (1) or False (0) instead, predicting something that is continuous like size.

• It has an S-shaped line.

• We can take our Linear Regression Model and convert it into Logistic Regression model with the help of Sigmoid Function.

• Logistic Regression's ability to provide probabilities and classify new samples using continuous and discrete measurements makes it a popular machine learning method.

- **Advantages and Disadvantages**

Advantages

- It doesn't require high computational power.
- Is easily interpretable.
- Is used widely by the data analyst and data scientists.
- Is very easy to implement.
- It doesn't require scaling of features.
- It provides a probability score for observations.

Disadvantages

- While working with Logistic regression you are not able to handle a large number of categorical features/variables.
- It is vulnerable to overfitting.
- It can't solve the non-linear problem with the logistic regression model that is why it requires a transformation of non-linear features.
- Logistic regression will not perform well with independent(X) variables that are not correlated to the target(Y) variable

- **K Nearest Neighbor (KNN)**

K nearest neighbors or KNN Algorithm is a simple algorithm which uses the entire dataset in its training phase. Whenever a prediction is required for an unseen data instance, it searches through the entire training dataset for k-most similar instances and the data with the most similar instance is finally returned as the prediction. KNN is often used in search applications where you are looking for similar items, like find items similar to this one.

KNN is non-parametric, meaning it does not make assumptions about the underlying data distribution, and lazy, as it does not require a training phase where it learns a model. Instead, KNN stores the entire training dataset and computes predictions at runtime based on the distances between data points, making it memory-intensive but flexible. KNN excels inscenarios where the decision boundary is highly irregular or the data distribution is complex. However, it may suffer from computational inefficiency and sensitivity to irrelevant or redundant features, necessitating careful preprocessing. KNN remains a popular choice for its simplicity, interpretability, and ability to handle non- linear relationshipsin data.

- **Features of KNN Algorithm**
  - KNN is a Supervised Learning algorithm that uses labelled input data set to predict the output of the data points.
  - It is one of the simplest Machine learning algorithms and it can be easily implemented for a varied set of problems.
  - It is mainly based on feature similarity. KNN checks how similar a data point is to its neighbor and classifies the data point into the class it is most similar to.
  - Unlike most algorithms, KNN is a non-parametric model which means that it does not make any assumptions about the data set. This makes the algorithm more effective since it can handle realistic data.
  - KNN is a lazy algorithm; this means that it memorizes the training data set instead of learning a discriminative function from the training data.
  - KNN can be used for solving both classification and regression problems.
- **Advantages and Disadvantages**

  Advantages
  - The algorithm is simple and easy to implement.
  - There's no need to build a model, tune several parameters, or make additional assumptions.
  - The algorithm is versatile. It can be used for classification, regression, and search.
  - The training phase of K-nearest neighbor classification is much faster compared to other classification algorithms. There is no need to train a model for generalization, that is why KNN is known as the simple and instance-based learning algorithm.
  - KNN can be useful in case of nonlinear data. It can be used with the regression problem. Output value for the object is computed by the average of k closest neighbors value.

  Disadvantages
  - The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.
  - The testing phase of K-nearest neighbor classification is slower and costlier in terms of time and memory. It requires large memory for storing the entire training dataset for prediction.
  - KNN requires scaling of data because KNN uses the Euclidean distance between two

data points to find nearest neighbors. Euclidean distance is sensitive to magnitudes. The features with high magnitudes will weigh more than features with low magnitudes.

- KNN also not suitable for large dimensional data.

- **Support Vector Machine (SVM)**

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification and regression challenges. Fig 2.10 represents the plot of ideal SVM algorithm. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n- dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate.

- SVM is a Supervised Learning algorithm that uses labelled input data set to predictthe output of the data points.
- It is one of the simplest Machine learning algorithms and it can be easily implementedfor a varied set of problems.
- SVM can be used for solving both classification and regression problems.

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. So,as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes.

The SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectorsand the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The**hyperplane** with maximum margin is called the **optimal hyperplane**.

SVM works relatively well when there is a clear margin of separation between classes. It is more effective in high dimensional spaces. SVM is effective in cases where the number of dimensions is greater than the number of samples and it is relatively memory efficient. SVM algorithm is not suitable for large data sets. It does not perform very well when each data point exceeds the number of training data samples, the SVM will underperform. As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification. The key idea behind SVM is to transform the input data into a higher-dimensional space where it becomes linearly separable, allowing for the creation of a hyperplane that effectively separates the classes.

Decision trees are versatile and intuitive machine learning models used for both classification and regression tasks. Structured as a hierarchical tree-like
structure, decision trees partition the feature space into segments, making decisions based on simple rules inferred from the data. At each internal node of the tree, a decision is made based on a feature's value, leading to different branches corresponding to different outcomes.

These decisions ultimately lead to leaf nodes, where the final prediction or decision is made. Decision trees are favored for their interpretability, as they provide clear insights into the decision-making process. In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree.

- **Features of SVM**
    - SVM is a Supervised Learning algorithm that uses labelled input data set to predict the output of the data points.
    - It is one of the simplest Machine learning algorithms and it can be easily implemented for a varied set of problems.
    - SVM can be used for solving both classification and regression problems.
- **Advantages and Disadvantages**

  Advantages
    - SVM works relatively well when there is a clear margin of separation between classes.
    - SVM is more effective in high dimensional spaces.
    - SVM is effective in cases where the number of dimensions is greater than the number of samples.
    - SVM is relatively memory efficient.

  Disadvantages
    - SVM algorithm is not suitable for large data sets.
    - SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
    - In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.

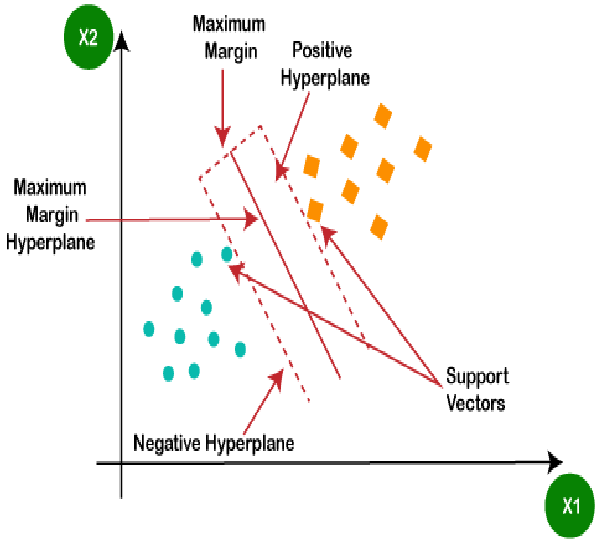


Fig 2.10: Plot of ideal SVM Algorithm

- **Decision Tree**

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.** It is a graphical representation for getting all the possible solutions to a problem/decision on based on given conditions. In a Decision tree, there are two nodes, which are the **Decision Node and Leaf Node**. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches as shown in Fig 2.11.

- **Features of Decision tree**
  - Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
  - The logic behind the decision tree can be easily understood because it shows a tree-like structure.
  - It is very easy to understand and implement.
- **Working**

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step3.

Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

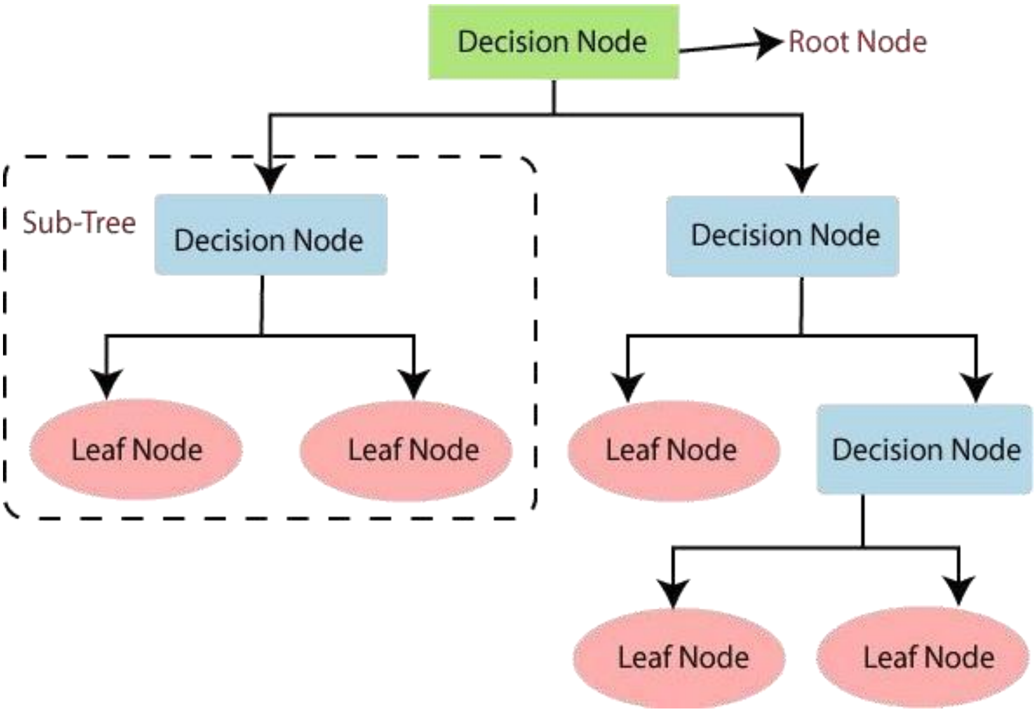


Fig 2.11: Ideal diagram of a Decision Tree

- A decision tree is a flowchart-like model that uses a branching approach to classify or predict something.
- The tree consists of internal nodes, representing the tests or decisions made, and leaf nodes, which represent the classifications or predictions.
- In the image, the root node represents the initial question or decision.
- The decision tree makes decisions based on the features of a data point.

- For instance, a decision tree to classify emails as spam or not spam might consider features like the sender's address, the subject line, and the presence of certain keywords.

- **Advantages and Disadvantages**

  Advantages

  - It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
  - It can be very useful for solving decision-related problems. It helps to think about all the possible outcomes for a problem.
  - Certain decision tree algorithms can handle problems with multiple outputs, predicting more than one value at a time.
  - Decision trees can work effectively with categorical data, which exists in distinct categories rather than continuous values. This makes them versatile for various datasets.
  - The tree structure inherently performs a form of feature selection by identifying the most important features for making predictions.
  - There is less requirement of data cleaning compared to other algorithms.

  Disadvantages

  - The decision tree contains lots of layers, which makes it complex. It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
  - For more class labels, the computational complexity of the decision tree may increase.
  - Decision trees can be prone to overfitting, especially when the tree grows too deep or when the dataset is noisy. Deep trees with many branches tend to capture noise in the training data, leading to poor generalization performance on unseen data.
  - Decision trees are sensitive to small variations in the training data. Even a slight change in the dataset or feature values can lead to a completely different tree structure, making decision trees less stable compared to other algorithms like linear models.
  - Decision trees have a tendency to create biased trees when the classes in the dataset are imbalanced. In such cases, the tree may be biased towards the majority class, leading to poor predictive performance for the minority class.
  - Decision trees may struggle to capture complex relationships or interactions between features, particularly when the decision boundaries are highly irregular or require multiple splits to delineate.
  - Decision trees are inherently greedy algorithms, meaning they make locally optimal decisions at each node without considering the global optimal tree structure.

Decision trees stand out in the world of machine learning for their unique ability to be understood by humans. Their branching structure, resembling a flowchart, allows you to see the thought process behind the model's predictions. This level of interpretability is invaluable for explaining the model's logic and gaining insights from the data itself. Additionally, decision trees are relatively simple to learn and implement, making them a good starting point for beginners or projects where clear explanations are essential. Beyond interpretability, they offer practical advantages.

2.4 The Most Challenging Task Performed

During my internship in artificial intelligence and machine learning, I encountered a particularly demanding task involving the classification of migraine episodes using K-Nearest Neighbors (KNN). This endeavor demanded a deep understanding of the complex patterns and triggers associated with migraine attacks. Fine-tuning the KNN algorithm to effectively classify migraine episodes based on their symptoms, duration, and severity presented a formidable challenge, requiring careful consideration of factors like feature selection, distance metrics, and the optimal number of neighbors. Simultaneously, in the AI project focused on classifying Ambassador cars and SUV cars, I faced unique hurdles in accurately distinguishing between these two iconic automobile categories. Gathering comprehensive data encompassing various models of Ambassador cars and SUVs, along with their distinguishing features such as engine specifications, dimensions, and historical performance, posed a significant obstacle.

Overcoming these challenges in both projects necessitated a multidisciplinary approach. Collaborating with neurologists and migraine specialists provided invaluable insights into the factors influencing migraine occurrences, facilitating feature selection and model refinement for migraine classification. Similarly, engaging with automotive experts and enthusiasts aided in acquiring relevant data and defining distinct classification criteria for Ambassador cars and SUVs. Technical proficiency in data preprocessing, model development, and evaluation played a pivotal role in both endeavors.

2.5 Problem Statement

Migraine Prediction using K Nearest Neighbors:

- Training the Model based on the data collected and processed.
- Developing a machine learning model using K Nearest Neighbors to predict the likelihood of migraine episodes based on various input factors.
- The K Nearest Neighbors algorithm can serve as a predictive tool to estimate the probability of a migraine attack by analyzing similarities between the individual's features and those of individuals who have experienced migraine episodes and those who have not in the training dataset.
- By training a K Nearest Neighbors model on a dataset containing labeled instances of individuals with and without migraine episodes, the algorithm learns to classify new individuals into either category based on the proximity of their features to the nearest neighbors in the feature space.
- This predictive model can assist healthcare professionals and individuals prone to migraines in identifying potential triggers and warning signs of migraine episodes, enabling proactive measures to mitigate the impact and severity of migraines.
- Furthermore, the K Nearest Neighbors model can provide insights into the key factors influencing migraine occurrences, offering valuable information for personalized migraine management strategies and preventive measures tailored to individual needs and lifestyle factors.

Classify Ambassador Car and SUV Car Classification in Images:

- Create an artificial intelligence system capable of accurately distinguishing between images of Ambassador and SUV car.
- The project involves training to recognize key visual features that differentiate between the two types of cars.
- These algorithms analyze various features of the vehicles, such as engine power, size, shape, and historical data, to categorize them accurately. For instance, SUVs typically have higher ground clearance and larger cargo space compared to sedans like the Ambassador.
- By training AI models on labeled datasets, we can develop robust classification systems that automatically distinguish between these two types of vehicles with high accuracy, aiding in inventory management, market analysis, and customer recommendation systems for automotive companies.

CHAPTER 3

REFLECTIONS

3.1 Prediction of Migraine Classification using K Nearest Neighbors

Age	Duration	Frequenc	Location	Characte	Intensity	Nausea	Vomit	Phonoph	Photophc	Visual	Sensory	Dysphasi	Dysarthri	Vertigo	Tinnitus	Hypoacus	Diplopia	Defect	Ataxia	Conscien	Paresthe	DPF	Type
30	1	5	1	1	2	1	0	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	Typical aura with migraine
50	3	5	1	1	3	1	1	1	1	2	1	0	0	1	0	0	0	0	0	0	0	0	Typical aura with migraine
53	2	1	1	1	2	1	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	Typical aura with migraine
45	3	5	1	1	3	1	0	1	1	2	2	0	0	1	0	0	0	0	0	0	0	0	Typical aura with migraine
53	1	1	1	1	2	1	0	1	1	4	0	0	0	0	0	0	0	0	0	0	0	1	Typical aura with migraine
49	1	1	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
27	1	5	1	1	3	1	0	1	1	2	0	0	0	1	1	0	0	0	0	0	0	0	Basilar-type aura
24	1	1	1	1	2	1	0	1	1	2	2	0	0	1	0	0	0	0	0	0	0	1	Typical aura with migraine
50	1	5	1	1	2	1	1	1	1	2	2	0	0	1	0	0	0	0	0	0	0	1	Typical aura with migraine
23	1	1	1	1	3	1	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	Typical aura with migraine
48	1	2	1	1	3	1	1	1	1	3	2	0	0	0	0	0	0	0	0	0	0	0	Typical aura with migraine
51	3	1	1	1	3	1	0	1	1	2	1	0	0	0	0	0	0	0	0	0	0	1	Typical aura with migraine
49	2	5	1	1	3	1	0	1	1	3	0	0	0	0	0	0	0	0	0	0	0	1	Typical aura with migraine
34	1	1	1	1	3	1	0	1	1	2	0	0	0	0	0	0	0	0	0	0	0	0	Typical aura with migraine
20	3	5	1	1	3	1	0	1	1	2	0	0	0	0	0	0	0	0	0	0	0	1	Typical aura with migraine
53	3	5	1	1	3	1	0	1	1	2	0	0	0	1	0	0	0	0	0	0	0	1	Typical aura with migraine
40	3	1	1	1	3	1	0	1	1	4	0	0	0	1	0	0	0	0	0	0	0	1	Typical aura with migraine
56	1	1	1	1	3	1	1	1	1	2	0	0	0	0	0	0	0	0	0	0	0	1	Typical aura with migraine
44	3	5	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
20	3	8	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
46	1	5	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
25	3	7	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
38	1	5	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
35	2	5	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
17	1	6	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
36	2	5	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
31	1	7	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
67	3	5	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
17	1	5	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
46	1	5	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
51	3	7	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
17	3	6	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
22	3	6	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
48	2	4	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
22	2	3	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
68	2	3	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
43	3	5	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
17	2	5	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
34	2	5	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
41	3	7	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
37	1	5	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura
26	2	3	1	1	3	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	Migraine without aura
24	1	4	1	1	3	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	Migraine without aura

Fig 3.1: Dataset

Fig 3.1 table containing data about patient details. The table includes the following columns:

- Age: This appears to be the age of the person who experienced the migraine.
- Duration: This is the duration of the migraine in hours.
- Frequency: This is the frequency of the migraine, indicated by a number but it doesn't specify the time unit (e.g., days, weeks, months).
- Location: This is the location of the pain in the head. It appears to be coded, with "1" indicating left.

- Intensity: This is the intensity of the pain on a scale of 0 to 4, with 0 being no pain and 4 being the worst pain imaginable.
- Nausea: This is a binary field indicating whether or not the person experienced nausea.
- Phonophobia: This is a binary field indicating whether or not the person experienced phonophobia (sensitivity to sound) (1 for yes, 0 for no).
- Dysphasia: This is a binary field indicating whether or not the person experienced dysphasia (difficulty speaking) (1 for yes, 0 for no).
- Dysarthria: This is a binary field indicating whether or not the person experienced dysarthria (difficulty slurring words) (1 for yes, 0 for no).
- Vertigo: This is a binary field indicating whether or not the person experienced vertigo (dizziness) (1 for yes, 0 for no).

```
def getPredictions(Age,Duration,Frequency,Location,Character,Intensity,Nausea,Vomit,Phonophobia,Photophobia,Visual,Sensory,Dysphasia,Dysarthria,Vertigo,Tinnitus,Hypoacusi
    data = pd.read_csv('C:\\Users\\lenovo\\Desktop\\Suryansh_ML\\models\\data.csv')
    import sklearn
    from sklearn.preprocessing import LabelEncoder
    le = LabelEncoder()
    data['Type_n'] = le.fit_transform(data['Type'])
    import sklearn
    X=data.drop(['Type','Type_n'],axis=1)
    y=data['Type_n']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Standardize the features (important for KNN)
    scaler = StandardScaler()
    X_train = scaler.fit_transform(X_train)
    X_test = scaler.transform(X_test)

    # Create a KNN classifier
    k = 8  # You can choose the value of k based on your requirement
    knn_classifier = KNeighborsClassifier(n_neighbors=k)

    # Train the classifier
    knn_classifier.fit(X_train, y_train)

    # Make predictions on the test set
    y_pred = knn_classifier.predict(X_test)

    # Evaluate the model
    accuracy = accuracy_score(y_test, y_pred)
    classification_report_str = classification_report(y_test, y_pred)

    # Print the results
    # print(f'Accuracy: {accuracy:.2f}')
    # print('Classification Report:\n', classification_report_str)
    prediction = knn_classifier.predict([
        [Age,Duration,Frequency,Location,Character,Intensity,Nausea,Vomit,Phonophobia,Photophobia,Visual,Sensory,Dysphasia,Dysarthria,Vertigo,Tinnitus,Hypoacusis,Diplopia
```

Fig 3.2: Code Snippet

Fig 3.2 represents the code snippet seems to be training a K-Nearest Neighbors (KNN) classification model to predict patient migraines. In this case, the model is likely trying to predict the types of migraines from the training data.

```
Accuracy: 0.89
Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.67      0.80         6
           1       0.75      1.00      0.86         3
           2       0.86      0.92      0.89        13
           3       1.00      1.00      1.00         4
           4       0.00      0.00      0.00         2
           5       0.90      0.92      0.91        49
           6       1.00      1.00      1.00         3

    accuracy                           0.89        80
   macro avg       0.79      0.79      0.78        80
weighted avg       0.88      0.89      0.88        80

Confusion Matrix:
 [[ 4  0  0  0  0  2  0]
 [ 0  3  0  0  0  0  0]
 [ 0  0 12  0  0  1  0]
 [ 0  0  0  4  0  0  0]
 [ 0  0  0  0  0  2  0]
 [ 0  1  2  0  1 45  0]
 [ 0  0  0  0  0  0  3]]
```

Fig 3.3: Confusion Matrix

The confusion matrix is a table that shows the number of correct and incorrect predictions made by the model. In this case, the model is trying to predict the type of migraine. The Fig.3.3 sent is showing the results of a machine learning classification task. The table shows the accuracy, precision, recall, F1-score, and support for each class, as well as the confusion matrix.

- Accuracy: 0.89
- Precision: This is a measure of how many times a model correctly predicted a positive case. A high precision means that most of the positive predictions were correct.
- Recall: This is a measure of how many times the model correctly identified all positive cases. A high recall means that the model didn't miss many positive cases.
- F1-Score: This is a harmonic mean of precision and recall. A high F1-score means that the model balanced both precision and recall well.
- Support: This is the number of data points for each class.

The confusion matrix shows that the model is more accurate at predicting type of migraines. It is important to consider the trade-off between precision and recall when evaluating a classification model. A high precision means that the model is good at not making false positive predictions, but a low recall means that the model is missing some of the true positive cases.

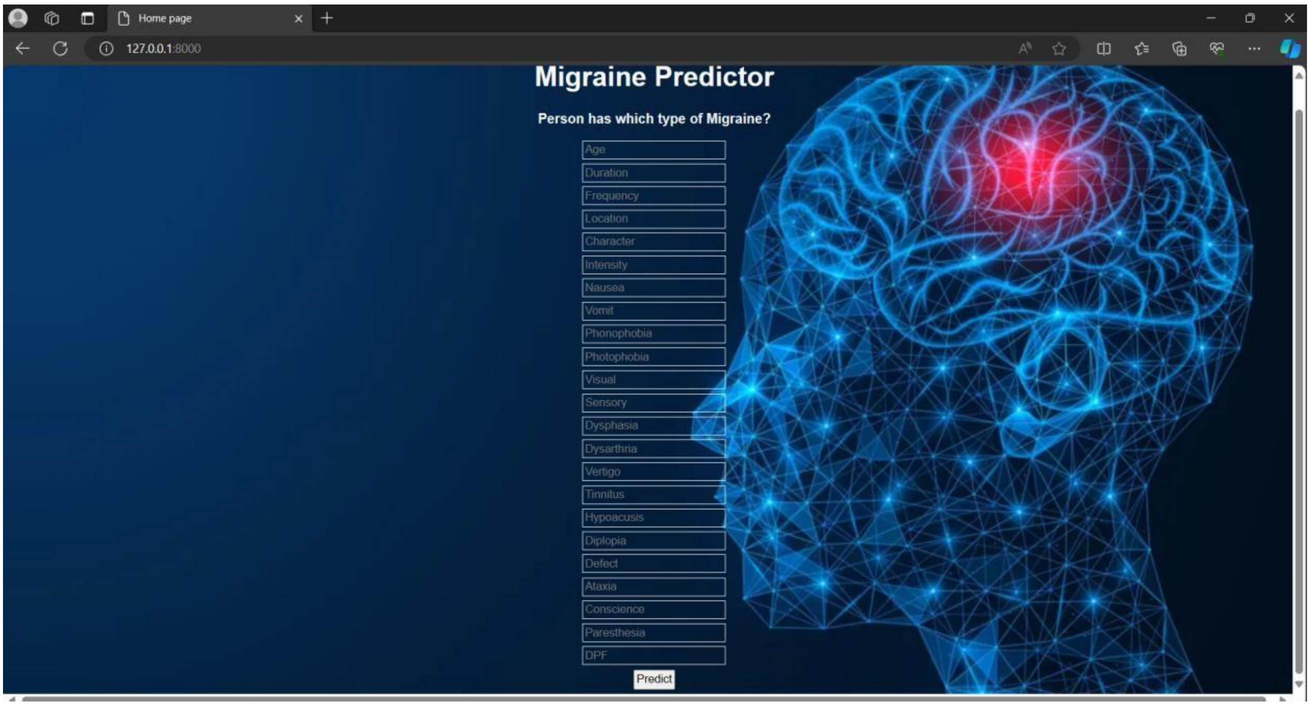


Fig 3.4: Home Page Snapshot

The dataset used in this project consists of many variables: Age, Duration, Frequency, Location, Intensity, Nausea, Vomit, Phonophobia, Photophobia, Visual, Sensory, Dysphasia, Dysarthria, Vertigo, Tinnitus, Hypoacusis, Diplopia, Defect, Ataxia, Consciousness, Paresthesia, DPF, Type inputs given as shown in Fig 3.4.

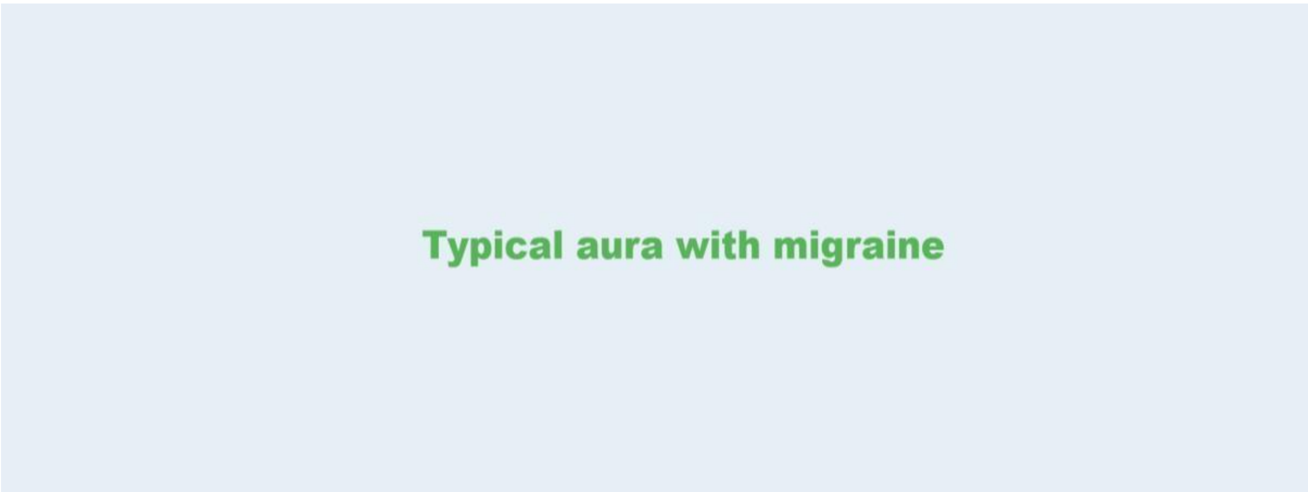


Fig 3.5: Result Snapshot

Based on the input parameters provided in the Fig 3.4 we developed a model that has the capacity of predicting fracture by making use of the information provided in data.csv Dataset as shown in Fig 3.5. By analyzing information within the data.csv dataset, we developed a migraines prediction model.

3.2 Classification of Ambassador Car and SUV Car

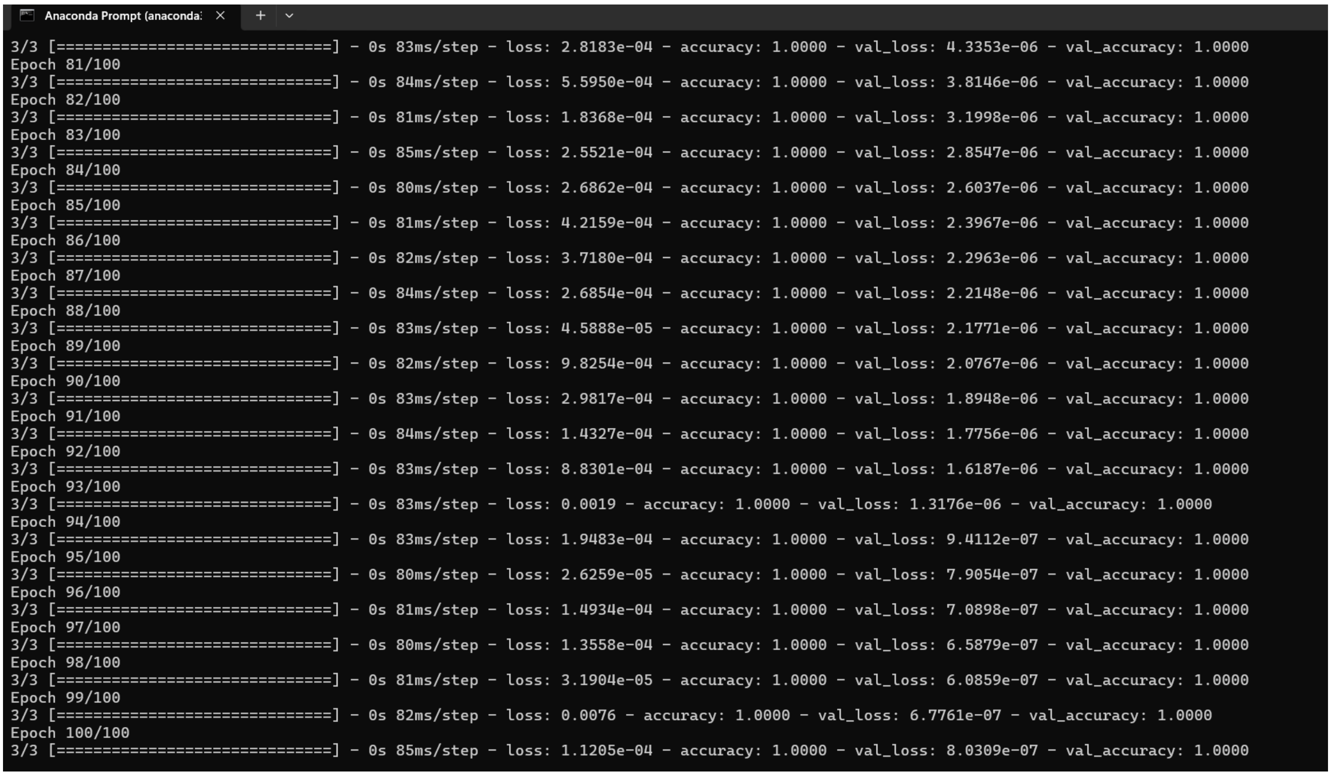
```python
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
import os

data = []
labels = []
classes = 2
cur_path = os.getcwd() #To get current directory


classs = { 1:"AmbassadorCar",
    2:"SUVCar"
}
i=1
#Retrieving the images and their labels
print("Obtaining Images & its Labels..............")
for i in range(classes):
    path = os.path.join(cur_path,'my_dir/train/',str(i))
    images = os.listdir(path)

    for a in images:
        try:
            image = Image.open(path + '\\'+ a)
            image = image.resize((30,30))
            image = np.array(image)
            data.append(image)
            labels.append(i)
            #print("{0} Loaded".format(a))
        except:
            print("Error loading image")
print("Dataset Loaded")
```

Fig 3.6: Code Snippet of Training

This Python code utilizes libraries like TensorFlow and Keras to train a Convolutional Neural Network (CNN) for classifying images of Suv and Ambassador car as shown in fig 3.6.

Firstly, the code prepares the data. It defines folders containing images of different types of Suv and Ambassador car images and loads them. Images are then resized and converted to a format suitable for the model. Next, the data is split into training and testing sets, and labels are converted to a one-hot encoded format for better handling by the machine learning model.

The core part of the code involves building and training the CNN model. The model uses convolutional layers to extract features from the images, pooling layers to reduce data size, and dropout layers to prevent overfitting. Finally, the model predicts probabilities for each image belonging to a specific air cooler class. After training the model on the prepared data, the code saves it for future use. This allows you to use the trained model to classify new air cooler images.



```
Anaconda Prompt (anaconda3)
3/3 [==============================] - 0s 83ms/step - loss: 2.8183e-04 - accuracy: 1.0000 - val_loss: 4.3353e-06 - val_accuracy: 1.0000
Epoch 81/100
3/3 [==============================] - 0s 84ms/step - loss: 5.5950e-04 - accuracy: 1.0000 - val_loss: 3.8146e-06 - val_accuracy: 1.0000
Epoch 82/100
3/3 [==============================] - 0s 81ms/step - loss: 1.8368e-04 - accuracy: 1.0000 - val_loss: 3.1998e-06 - val_accuracy: 1.0000
Epoch 83/100
3/3 [==============================] - 0s 85ms/step - loss: 2.5521e-04 - accuracy: 1.0000 - val_loss: 2.8547e-06 - val_accuracy: 1.0000
Epoch 84/100
3/3 [==============================] - 0s 80ms/step - loss: 2.6862e-04 - accuracy: 1.0000 - val_loss: 2.6037e-06 - val_accuracy: 1.0000
Epoch 85/100
3/3 [==============================] - 0s 81ms/step - loss: 4.2159e-04 - accuracy: 1.0000 - val_loss: 2.3967e-06 - val_accuracy: 1.0000
Epoch 86/100
3/3 [==============================] - 0s 82ms/step - loss: 3.7180e-04 - accuracy: 1.0000 - val_loss: 2.2963e-06 - val_accuracy: 1.0000
Epoch 87/100
3/3 [==============================] - 0s 84ms/step - loss: 2.6854e-04 - accuracy: 1.0000 - val_loss: 2.2148e-06 - val_accuracy: 1.0000
Epoch 88/100
3/3 [==============================] - 0s 83ms/step - loss: 4.5888e-05 - accuracy: 1.0000 - val_loss: 2.1771e-06 - val_accuracy: 1.0000
Epoch 89/100
3/3 [==============================] - 0s 82ms/step - loss: 9.8254e-04 - accuracy: 1.0000 - val_loss: 2.0767e-06 - val_accuracy: 1.0000
Epoch 90/100
3/3 [==============================] - 0s 83ms/step - loss: 2.9817e-04 - accuracy: 1.0000 - val_loss: 1.8948e-06 - val_accuracy: 1.0000
Epoch 91/100
3/3 [==============================] - 0s 84ms/step - loss: 1.4327e-04 - accuracy: 1.0000 - val_loss: 1.7756e-06 - val_accuracy: 1.0000
Epoch 92/100
3/3 [==============================] - 0s 83ms/step - loss: 8.8301e-04 - accuracy: 1.0000 - val_loss: 1.6187e-06 - val_accuracy: 1.0000
Epoch 93/100
3/3 [==============================] - 0s 83ms/step - loss: 0.0019 - accuracy: 1.0000 - val_loss: 1.3176e-06 - val_accuracy: 1.0000
Epoch 94/100
3/3 [==============================] - 0s 83ms/step - loss: 1.9483e-04 - accuracy: 1.0000 - val_loss: 9.4112e-07 - val_accuracy: 1.0000
Epoch 95/100
3/3 [==============================] - 0s 80ms/step - loss: 2.6259e-05 - accuracy: 1.0000 - val_loss: 7.9054e-07 - val_accuracy: 1.0000
Epoch 96/100
3/3 [==============================] - 0s 81ms/step - loss: 1.4934e-04 - accuracy: 1.0000 - val_loss: 7.0898e-07 - val_accuracy: 1.0000
Epoch 97/100
3/3 [==============================] - 0s 80ms/step - loss: 1.3558e-04 - accuracy: 1.0000 - val_loss: 6.5879e-07 - val_accuracy: 1.0000
Epoch 98/100
3/3 [==============================] - 0s 81ms/step - loss: 3.1904e-05 - accuracy: 1.0000 - val_loss: 6.0859e-07 - val_accuracy: 1.0000
Epoch 99/100
3/3 [==============================] - 0s 82ms/step - loss: 0.0076 - accuracy: 1.0000 - val_loss: 6.7761e-07 - val_accuracy: 1.0000
Epoch 100/100
3/3 [==============================] - 0s 85ms/step - loss: 1.1205e-04 - accuracy: 1.0000 - val_loss: 8.0309e-07 - val_accuracy: 1.0000
```

Fig 3.7: Training the Data

The Fig. 3.7 is a screenshot of a command line window, likely showing the output of running the Python training code. An epoch refers to one complete pass through the entire training dataset. Here's an explanation of some of the elements seen in each line:

- **Epoch [number]**: This specifies the current epoch number (e.g., Epoch 81).
- **Os/Bs [time]**: This indicates the operating system (Os) or batch size (Bs) followed by the time taken per step in the epoch.
- **loss**: This refers to the loss function, a measure of how well the model's predictions deviate from the ground truth. Lower loss indicates better performance.
- **accuracy**: This represents the model's accuracy, the proportion of correct predictions made on a set of data.
- **val loss/val_accuracy**: These indicate the loss and accuracy on the validation set, a portion of the data used to monitor the model's generalization ability during training.

The values you see here suggest that the model is performing well with a very low loss value and high accuracy on both the training and validation sets. As the epochs progress, the loss decreases and the accuracy increases, signifying that the model is learning to classify different types of Suv and Ambassador car images accurately.

```python
from tensorflow.keras.models import Sequential, load_model
from tensorflow.keras.layers import Conv2D, MaxPool2D, Dense, Flatten, Dropout
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
import os

data = []
labels = []
classes = 2
cur_path = os.getcwd() #To get current directory


classs = { 1:"AmbassadorCar",
    2:"SUVCar"
}
i=1
#Retrieving the images and their labels
print("Obtaining Images & its Labels..............")
for i in range(classes):
    path = os.path.join(cur_path,'my_dir/train/',str(i))
    images = os.listdir(path)

    for a in images:
        try:
            image = Image.open(path + '\\'+ a)
            image = image.resize((30,30))
            image = np.array(image)
            data.append(image)
            labels.append(i)
            #print("{0} Loaded".format(a))
        except:
            print("Error loading image")
print("Dataset Loaded")
```

Fig 3.8: Code Snippet of Testing

This Python code tackles classifying different types of Suv and Ambassador car images. It first prepares the data by loading images of different objects from folders, resizing them, and converting them into a format usable by a machine learning model. Labels are assigned based on the brand. The data is then split into training and testing sets as shown in Fig 3.8.

Next, it assumes a pre-trained model named "my_model.h5" exists, which likely learned to distinguish between air cooler images during training. A function called "classify" takes an image file path and uses the loaded model to predict the Suv or car in the image.

Finally, the script identifies image files in a test directory and uses the "classify" function on each image, printing the predicted object based on a predefined mapping. This allows you to test how well the pre-trained model generalizes to unseen images. After testing the model on the prepared data, the code saves it for future use. This allows you to use the tested model to classify new Suv and car images.

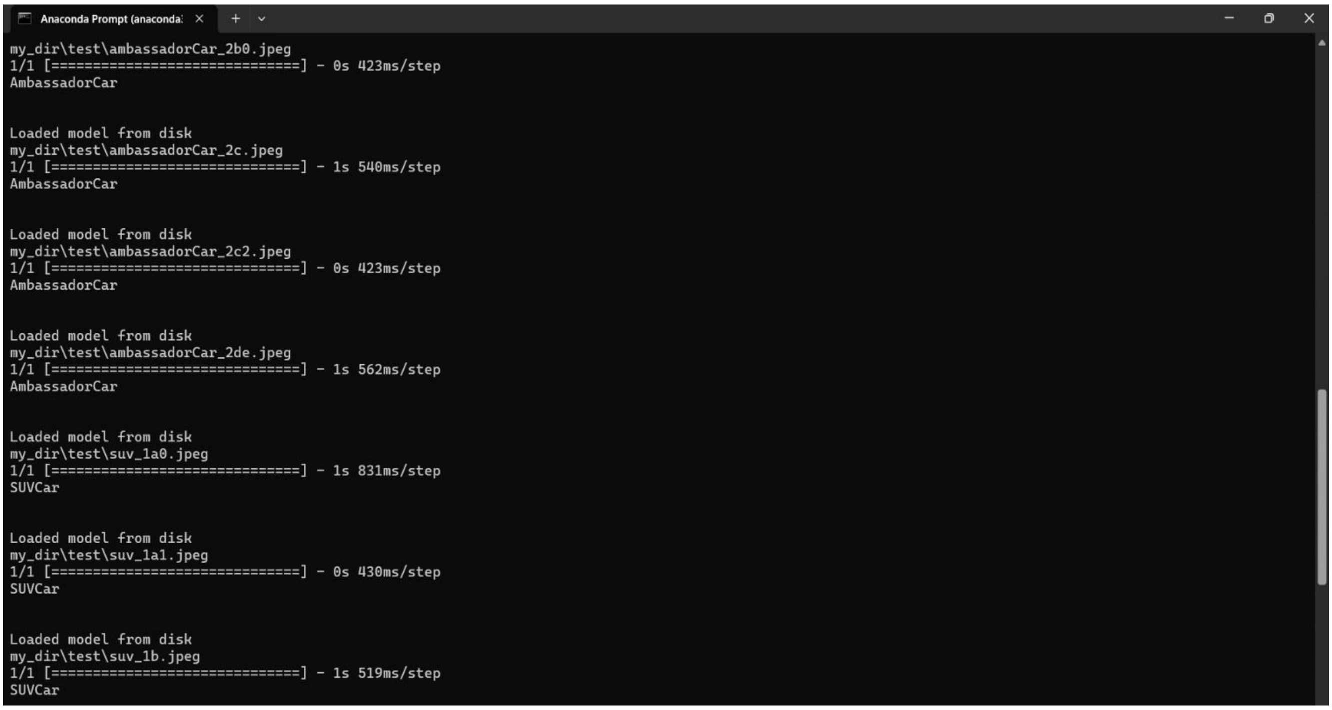


Fig 3.9: Testing the Data

The Fig 3.9 depicts the successful execution of testing the data and image classification of suv and ambassador car. The script appears to be loading images from a folder named "my_dir/test". Each image is then fed into a machine learning model, which has been loaded successfully according to the message "Loaded model from disk". This model classifies the images, and the results are displayed alongside the processing time for each image. For instance, one image named "suv_1a1.jpeg" was classified as a "SUVCar" and so other as “AmbassadorCar”.In summary, using the loaded model to successfully classify images for different types of suv and car images.

CHAPTER 4

CONCLUSION

In conclusion, the training on artificial intelligence and machine learning has provided us with a comprehensive understanding of how to prepare data, build models and evaluate their performance through hands-on exercises and projects. We have learnt the basics of supervised and unsupervised learning and explored various algorithms such as decision tree, random forest and neural networks.

It is important to remember that artificial intelligence and machine learning is a constantly evolving field and there is always more to learn. However, by understanding the fundamental concepts and techniques covered in the training, we will be well-equipped to tackle a wide range of machine learning problems. Furthermore, we have a solid foundation on which we can continue to build our knowledge and skills in artificial intelligence and machine learning.

CHAPTER 5

REFERENCES

BOOKS

[1] "The Hundred-Page Machine Learning Book" by Andriy Burkov.

[2] "Hands-On Machine Learning with Scikit-Learn and TensorFlow" by Aurélien Géron.

[3] "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville.

[4] "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili.

WEBSITES

- https://www.tensorflow.org/learn
- https://www.kaggle.com/learn/machine-learning
- https://machinelearningmastery.com/