

**TRAINING REPORT**

**OF**

**SUMMER TRAINING, UNDERTAKEN**

**AT**

**EXCELLENCE TECHNOLOGY**

**ON**

**SNAKE GAME**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE**

**OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**Computer Science and Engineering**

**Submitted By:**

**Name: Jaspreet Kaur**

**Roll No: 12101188**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**PUNJABI UNIVERSITY**

**PATIALA-14700**2

**CERTIFICATE**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**PUNJABI UNIVERSITY, PATIALA**

**CANDIDATE’S DECLARATION**

I, JASPREET KAUR, hereby declare that I have undertaken summer training at **EXCELLENCE TECHNOLOGY** during a period from JUNE to JULY in partial fulfilment of requirements for the award of degree of B. TECH in Computer Science & Engineering at Punjabi University, Patiala. The Project Work which is being presented in this training report is submitted to Department of Computer Science & Engineering at Punjabi University, Patiala is an authentic record of my work

Signature of Student: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The summer training Viva-Voce Examination of Jaspreet Kaur has been

held on SEPTEMBER 28,2023 and accepted.

Signature of Examiner:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ABSTRACT**

Games are the structured form of play, usually undertaken for entertainment or fun, and sometimes used for educational too. Many games are also considered to be work (such as professional players of spectator sports or games) or art (such as jigsaw puzzles or games involving an artistic layout such as Mahjong, Solitaire, or some video games).

Games are sometimes played purely for enjoyment, sometimes for achievement or reward as well. They can be played alone, in teams or online; by amateurs or by professionals. Key components of games are goals, rules and challenge and interaction. Games generally involve mental or physical stimulation, and often both. Many games help develop skills, serve as form of exercise and stimulation.

Snake game is one of the popular arcade games of all the times. In this game the main objective of the player is to catch maximum number of foods without hitting the wall or itself. Creating a snake game can be taken as challenge for learning Pygame in AI. It is one of the best beginner-friendly projects that every programmer should challenge.

I had used Pygame to create this Snake game. Pygame is an open-source library that is designed for making video games. It has inbuilt graphics and sound libraries. It is also beginner-friendly, and cross-platform.

**ACKNOWLEDGEMENT**

First of all, I would like to say that it was a great pleasure and privilege to have got the opportunity of undertaking a summer training at **EXCELLENCE TECHNOLOGY.**

I am thankful to **EXCELLENCE TECHNOLOGY** who is instrumental in creating an opportunity for undertake summer training in this esteemed organization. I am thankful to Mr. Dilip Singh who has always stood by our side and guided, appreciated and encouraged us to get more and more venture.

I am sure that knowledge and information that we have gained during this period would be of immense value in future.

**JASPREET KAUR**

**ABOUT THE COMPANY**

**EXCELLENCE TECHNOLOGY** (ET) is India based leading strategic IT Company offering integrated IT solutions with the vision to provide Excellence in software solution. EXCELLENCE TECHNOLOGY brings innovative ideas and cutting-edge technologies into Business of Customers. EXCELLENCE TECHNOLOGY is having a rich experience in providing high technology end to end solutions in MOBILE APP AND WEB DEVELOPMENT.

EXCLLENCE TECHNOLOGY managing global clients across various business verticals align IT strategies to achieve business goals. The various accreditations that they achieved for every service; they offer reflect our commitment towards the quality assurance. ET has won the NATIONAL AWARD in 2015-2016 for highly appreciable contribution in the field of computers from Honourable Education Minister of Punjab DR. Daljit Singh Cheema. Recently, ET has officially Registered in Canada under the Business Corporations Act on July10, 2023 in province of British Columbia, Canada.

**List of Figures**

|  |  |
| --- | --- |
| **Figure number** | **Page no.** |
| Figure 1.1 | 1 |
| Figure 1.2 | 2 |
| Figure 1.3 | 4 |
| Figure 1.4 | 5 |
| Figure 1.5 | 6 |
| Figure 1.6 | 8 |
| Figure 1.7 | 9 |
| Figure 1.8 | 11 |
| Figure 1.9 | 12 |
| Figure 2.1 | 28 |
| Figure 2.2 | 28 |
| Figure 2.3 | 29 |
| Figure 2.4 | 29 |
| Figure 2.5 | 30 |
| Figure 2.6 | 30 |
| Figure 2.7 | 31 |
| Figure 2.8 | 31 |

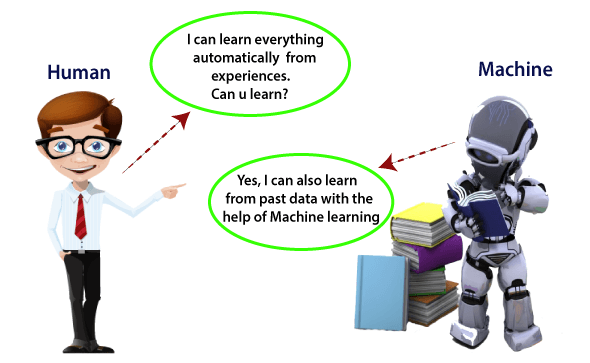
**Table of content**

|  |  |
| --- | --- |
| **Topics** | **Page no.** |
| Training Report | 1 |
| Company certificate | 2 |
| Candidate’s Declaration | 3 |
| Abstract | 4 |
| Acknowledgement | 5 |
| About the company | 6 |
| List of Figures | 7 |
| **Chapter 1 – Introduction** | 1-19 |
| 1.1 What is Machine learning | 1 |
| 1.2 How does Machine learning work | 1 |
| 1.3Features of Machine Learning | 2 |
| 1.4 History of Machine learning | 2 |
| 1.5 Types of Machine learning | 3-7 |
| 1.6 Challenges of Machine learning | 8-10 |
| 1.7 Application of Machine learning | 11-17 |
| 1.8 Future of Machine learning | 18 |
| 1.9 Jupyter | 19 |
| **CHAPTER 2 – Training work Undertaken** | 20-31 |
| 2.1 Title of Project | 21 |
| 2.2 History of Project | 21 |
| 2.3 Description of Game | 21 |
| 2.4 System Specification | 21 |
| 2.5 Pre-requistes of Project | 22 |
| 2.6 How are we going to build snake Game | 23 |
| 2.7 Steps to Build Snake Game | 23-27 |
| 2.8 Screenshots of Project | 28-31 |
| **CHAPTER 3 – Results** | 32 |
| 3.1 Result | 32 |
| 3.2 Discussion | 32 |
| **CHAPTER 4 – Conclusion and future scope** | 33 |
| 4.1 Conclusion | 33 |
| 4.2 Future scope | 33 |
| References | 33 |

**CHAPTER 1: INTRODUCTION**

**1.1 MACHINE LEARNING:**

Machine Learning is a type of artificial intelligence that allows software applications to become more accurate at predicting outcomes without being explicit programmed to do so. Machine Learning algorithms use historical data as input to predict new values.



*Figure1.1*

**1.2 How does Machine Learning work**

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:



*Figure 1.2*

**1.3 Features of Machine Learning:**

* Machine learning uses data to detect various patterns in a given dataset.
* It can learn from past data and improve automatically.
* It is a data-driven technology.

**1.4 History of Machine Learning**

This section discusses the development of machine learning over the years. Today we are witnessing some astounding applications like self-driving cars, natural language processing and facial recognition systems making use of ML techniques for their processing. All this began in the year 1943, when Warren McCulloch a neurophysiologist along with a mathematician named Walter Pitts authored a paper that threw a light on neurons and its working. They created a model with electrical circuits and thus neural network was born.

The famous “Turing Test” was created in 1950 by Alan Turing, which would ascertain whether computers had real intelligence. It has to make a human believe that it is not a computer but a human instead, to get through the test. Arthur Samuel developed the first computer program that could learn as it played the game of checkers in the year 1952. The first neural network, called the perceptron was designed by Frank Rosenblatt in the year 1957.

The big shift happened in the 1990s when machine learning moved from being knowledge-driven to a data-driven technique due to the availability of huge volumes of data. IBM’s Deep Blue, developed in 1997 was the first machine to defeat the world champion in the game of chess. Businesses have recognized that the potential for complex calculations could be increased through machine learning.  Some of the latest projects include: Google Brain, which was developed in 2012, was a deep neural network that focused on pattern recognition in images and videos. It was later employed to detect objects in YouTube videos. In 2014, Facebook created Deep Face, which can recognize people just like how humans do. In 2014, Deep Mind created a computer program called Alpha Go a board game that defeated a professional Go player. Due to its complexity, the game is said to be a very challenging yet classical game for artificial intelligence. Scientists Stephen Hawking and Stuart Russel have felt that if AI gains the power to redesign itself at an intensifying rate, then an unbeatable “intelligence explosion” may lead to human extinction. Musk characterizes AI as humanity’s “biggest existential threat.” Open AI is an organization created by Elon Musk in 2015 to develop safe and friendly AI that could benefit humanity. Recently, some of the breakthrough areas in AI are Computer Vision, Natural Language Processing and Reinforcement Learning.

**1.5 Why Should We Learn Machine Learning?**

Machine learning is a powerful tool that can be used to solve a wide range of problems. It allows computers to learn from data, without being explicitly programmed. This makes it possible to build systems that can automatically improve their performance over time by learning from their experiences.

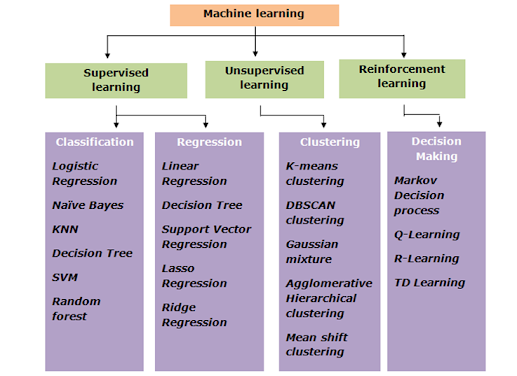
There are many reasons why learning machine learning is important:

* Machine learning is widely used in many industries, including healthcare, finance, and e-commerce. By learning machine learning, you can open up a wide range of career opportunities in these fields.
* Machine learning can be used to build intelligent systems that can make decisions and predictions based on data. This can help organizations make better decisions, improve their operations, and create new products and services.
* Machine learning is an important tool for data analysis and visualization. It allows you to extract insights and patterns from large datasets, which can be used to understand complex systems and make informed decisions.
* Machine learning is a rapidly growing field with many exciting developments and research opportunities. By learning machine learning, you can stay up-to-date with the latest research and developments in the field.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

**1.6 What are the different types of machine learning?**

There are four basic approaches: [supervised](https://www.techtarget.com/searchenterpriseai/definition/supervised-learning) learning, [unsupervised](https://www.techtarget.com/whatis/definition/unsupervised-learning) learning, semi- supervised learning and reinforcement learning.

****

*Figure1.3*

* **Supervised learning:** In this type of machine learning, [data scientists](https://www.techtarget.com/searchenterpriseai/definition/data-scientist) supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.
* **Unsupervised learning:** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.
* **Semi-supervised learning:** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled [training data](https://www.techtarget.com/searchenterpriseai/feature/Using-small-data-sets-for-machine-learning-models-sees-growth), but the model is free to explore the data on its own and develop its own understanding of the data set.
* **Reinforcement learning:**Data scientists typically use [reinforcement learning](https://www.techtarget.com/searchenterpriseai/definition/reinforcement-learning) to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

**1.6.1 How does supervised machine learning work?**

Supervised machine learning requires the [data scientist](https://www.techtarget.com/searchbusinessanalytics/feature/Key-differences-of-a-data-scientist-vs-data-engineer) to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks:

* **Binary classification:**Dividing data into two categories.
* **Multi-class classification:**Choosing between more than two types of answers.
* **Regression modeling:** Predicting continuous values.
* **Ensembling:** Combining the predictions of multiple machine learning models to produce an accurate prediction.



*Figure1.4*

**1.6.2 How does unsupervised machine learning work?**

Unsupervised machine learning algorithms do not require data to be labelled. They sift through unlabelled data to look for patterns that can be used to group data points into subsets. Most types of deep learning, including [neural networks](https://www.techtarget.com/searchenterpriseai/definition/neural-network), are unsupervised algorithms. Unsupervised learning algorithms are good for the following tasks:

* **Clustering:** Splitting the dataset into groups based on similarity.
* **Anomaly detection:** Identifying unusual data points in a data set.



*Figure1.5*

**1.6.3 How does semi-supervised learning work?**

Semi-supervised learning works by data scientists feeding a small amount of [labeled training data](https://www.techtarget.com/searchenterpriseai/feature/Labeled-data-brings-machine-learning-applications-to-life) to an algorithm. From this, the algorithm learns the dimensions of the data set, which it can then apply to new, unlabeled data. The performance of algorithms typically improves

* **Machine translation:** Teaching algorithms to translate language based on less than a full dictionary of words.
* **Fraud detection:** Identifying cases of fraud when you only have a few positive examples.
* **Labelling data:** Algorithms trained on small data sets can learn to [apply data labels](https://www.techtarget.com/whatis/definition/data-labeling) to larger sets automatically.

**1.6.4 How does reinforcement learning work?**

Reinforcement learning works by [programming an algorithm](https://www.techtarget.com/searchenterpriseai/feature/5-types-of-machine-learning-algorithms-you-should-know) with a distinct goal and a prescribed set of rules for accomplishing that goal. Data scientists also program the algorithm to seek positive rewards -- which it receives when it performs an action that is beneficial toward the ultimate goal -- and avoid punishments -- which it receives when it performs an action that gets it farther away from its ultimate goal. Reinforcement learning is often used in areas such as:

* **Robotics:** Robots can learn to perform tasks the physical world using this technique.
* **Video gameplay:** Reinforcement learning has been used to teach bots to play a number of video games.
* **Resource management:** Given finite resources and a defined goal, reinforcement learning can help enterprises plan out how to allocate resources.

**1.7 How to get started with Machine Learning?**

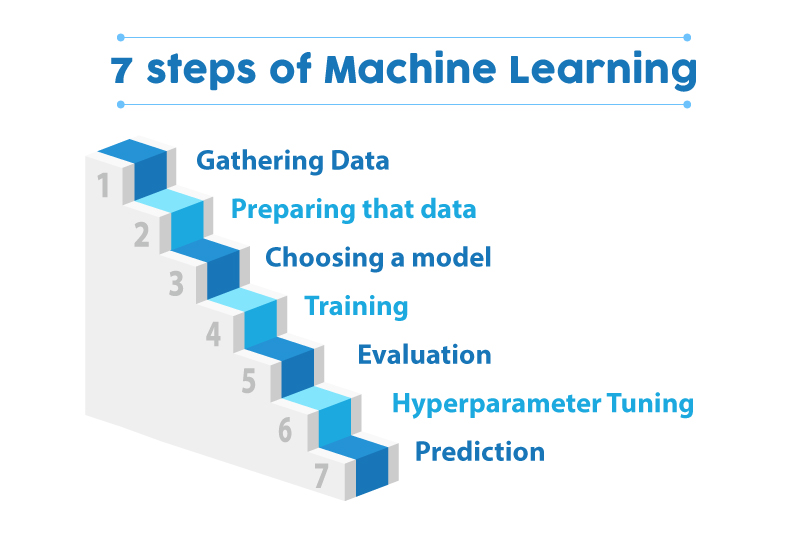
To get started, let’s take a look at some of the important terminologies.

**Terminology:**

* **Model**: Also known as “hypothesis”, a machine learning model is the mathematical representation of a real-world process. A machine learning algorithm along with the training data builds a machine learning model.
* **Feature**: A feature is a measurable property or parameter of the data-set.
* **Feature Vector**: It is a set of multiple numeric features. We use it as an input to the machine learning model for training and prediction purposes.
* **Training**: An algorithm takes a set of data known as “training data” as input. The learning algorithm finds patterns in the input data and trains the model for expected results (target). The output of the training process is the machine learning model.
* **Prediction**: Once the machine learning model is ready, it can be fed with input data to provide a predicted output.
* **Target (Label)**: The value that the machine learning model has to predict is called the target or label.
* **Overfitting**: When a massive amount of data trains a machine learning model, it tends to learn from the noise and inaccurate data entries. Here the model fails to characterize the data correctly.
* **Underfitting**: It is the scenario when the model fails to decipher the underlying trend in the input data. It destroys the accuracy of the machine learning model. In simple terms, the model or the algorithm does not fit the data well enough.

**1.9 There are Seven Steps of Machine Learning**

* Gathering Data
* Preparing that data
* Choosing a model
* Training
* Evaluation
* Hyperparameter Tuning
* Prediction



*Figure 1.6*

It is mandatory to learn a programming language, preferably Python, along with the required analytical and mathematical knowledge. Here are the five mathematical areas that you need to brush up before jumping into solving Machine Learning problems:

* Linear algebra for data analysis: Scalars, Vectors, Matrices, and Tensors
* Mathematical Analysis: Derivatives and Gradients
* Probability theory and statistics for Machine Learning
* Multivariate Calculus
* Algorithms and Complex Optimizations

**1.10 Blocks of Machine Learning**

The three major building blocks of a system are the model, the parameters, and the learner.

* Model is the system which makes predictions
* The parameters are the factors which are considered by the model to make predictions
* The learner makes the adjustments in the parameters and the model to align the predictions with the actual results

Let us build on the beer and wine example from above to understand how machine learning works. A machine learning model here has to predict if a drink is a beer or wine. The parameters selected are the color of the drink and the alcohol percentage. The first step is:

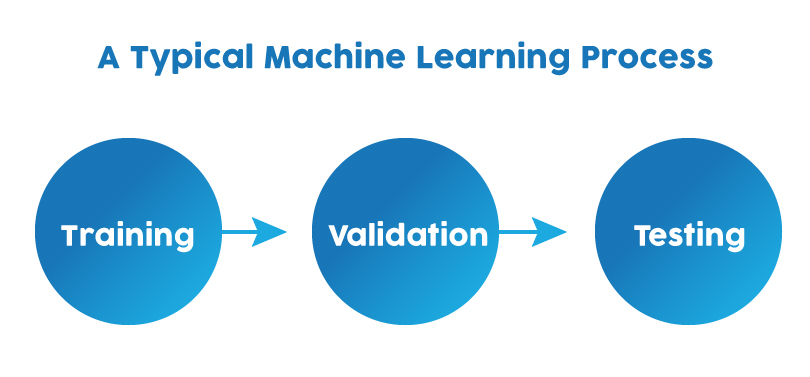
**1.8.1 Learning from the training set**

This involves taking a sample data set of several drinks for which the colour and alcohol percentage is specified. Now, we have to define the description of each classification, that is wine and beer, in terms of the value of parameters for each type. The model can use the description to decide if a new drink is a wine or beer.  
  
You can represent the values of the parameters, ‘colour’ and ‘alcohol percentages’ as ‘x’ and ‘y’ respectively. Then (x,y) defines the parameters of each drink in the training data. This set of data is called a training set. These values, when plotted on a graph, present a hypothesis in the form of a line, a rectangle, or a polynomial that fits best to the desired results.

**1.8.2 Measure error**

Once the model is trained on a defined training set, it needs to be checked for discrepancies and errors. We use a fresh set of data to accomplish this task. The outcome of this test would be one of these four:

* True Positive: When the model predicts the condition when it is present
* True Negative: When the model does not predict a condition when it is absent
* False Positive: When the model predicts a condition when it is absent
* False Negative: When the model does not predict a condition when it is present



*Figure 1.7*

The sum of FP and FN is the total error in the model.

**1.8.3 Manage Noise**

For the sake of simplicity, we have considered only two parameters to approach a machine learning problem here that is the colour and alcohol percentage. But in reality, you will have to consider hundreds of parameters and a broad set of learning data to solve a machine learning problem.

* The hypothesis then created will have a lot more errors because of the noise. Noise is the unwanted anomalies that disguise the underlying relationship in the data set and weakens the learning process. Various reasons for this noise to occur are:
* Large training data set
* Errors in input data
* Data labelling errors
* Unobservable attributes that might affect the classification but are not considered in the training set due to lack of data

You can accept a certain degree of training error due to noise to keep the hypothesis as simple as possible.

**1.8.4 Testing and Generalization**

While it is possible for an algorithm or hypothesis to fit well to a training set, it might fail when applied to another set of data outside of the training set. Therefore, It is essential to figure out if the algorithm is fit for new data. Testing it with a set of new data is the way to judge this. Also, generalisation refers to how well the model predicts outcomes for a new set of data.

When we fit a hypothesis algorithm for maximum possible simplicity, it might have less error for the training data, but might have more significant error while processing new data. We call this is underfitting. On the other hand, if the hypothesis is too complicated to accommodate the best fit to the training result, it might not generalise well. This is the case of over-fitting. In either case, the results are fed back to train the model further.

**1.9 Who's using machine learning and what's it used for?**

Today, machine learning is used in a wide range of applications. Perhaps one of the most well-known examples of machine learning in action is the [recommendation engine](https://www.techtarget.com/whatis/definition/recommendation-engine) that powers Facebook's news feed.

Facebook uses machine learning to personalize how each member's feed is delivered. If a member frequently stops to read a particular group's posts, the recommendation engine will start to show more of that group's activity earlier in the feed.

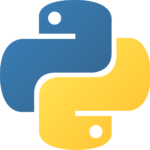
Behind the scenes, the engine is attempting to reinforce known patterns in the member's online behavior. Should the member change patterns and fail to read posts from that group in the coming weeks, the news feed will adjust accordingly.

**1.10 What is the future of machine learning?**

While machine learning algorithms have been around for decades, they've attained new popularity as [artificial intelligence](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence) has grown in prominence. Deep learning models, in particular, power today's most advanced AI applications.

Machine learning platforms are among enterprise technology's most competitive realms, with most major vendors, including Amazon, Google, Microsoft, IBM and others, racing to sign customers up for platform services that cover the spectrum of machine learning activities, including data collection, [data preparation](https://www.techtarget.com/searchbusinessanalytics/definition/data-preparation), data classification, model building, training and application deployment.

**1.11 Which Language is Best for Machine Learning?**



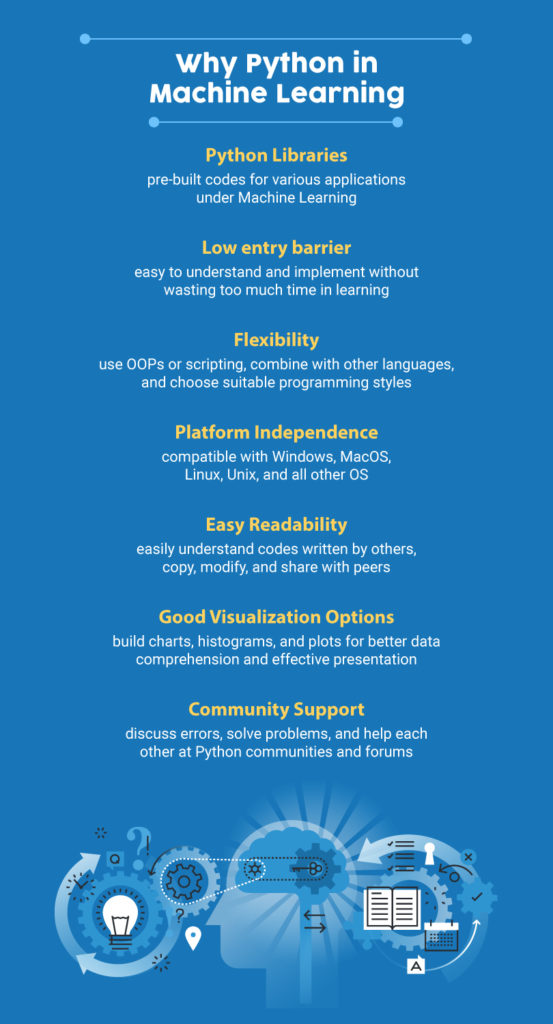
*Figure 1.8*

Python is hands down the best programming language for Machine Learning applications due to the various benefits mentioned in the section below. Other programming languages that could be used are: R, C++, JavaScript, Java, C#, Julia, Shell, TypeScript, and Scala.

Python is famous for its readability and relatively lower complexity as compared to other programming languages. ML applications involve complex concepts like calculus and linear algebra which take a lot of effort and time to implement. Python helps in reducing this burden with quick implementation for the ML engineer to validate an idea. You can check out the Python Tutorial to get a basic understanding of the language. Another benefit of using Python is the pre-built libraries. There are different packages for a different type of applications, as mentioned below:

* Numpy, OpenCV, and Scikit are used when working with images
* NLTK along with Numpy and Scikit again when working with text
* Librosa for audio applications
* Matplotlib, Seaborn, and Scikit for data representation
* TensorFlow and Pytorch for Deep Learning applications
* Scipy for Scientific Computing
* Django for integrating web applications
* Pandas for high-level data structures and analysis

Here is a summary:



*Figure 1.9*

|  |  |
| --- | --- |
| **Concept** | **Definition** |
| Artificial intelligence | The field of computer science aims to create intelligent machines that can think and function like humans. |
| Machine learning | A subfield of artificial intelligence that focuses on developing algorithms and models that can learn from data rather than being explicitly programmed. |
| Deep learning | A subfield of machine learning that uses multi-layered artificial neural networks to learn complex patterns in data. |

**1.12 Difference Between Machine Learning, Artificial intelligence and deep learning.**

*Table 1.1*

**1.13 Intelligence and Deep Learning**

Here is a brief summary of the main differences between these concepts:

* Artificial intelligence is a broad field that encompasses a variety of techniques and approaches for creating intelligent systems.
* The practice of teaching algorithms to learn from data rather than being explicitly programmed is known as machine learning, which is a subset of artificial intelligence.

Deep learning is a branch of machine learning that use multiple layers of artificial neural networks to discover intricate data patterns.

**1.14 Advantages and Disadvantages**

Everything comes with a few advantages and disadvantages. In this section, let’s talk about a few of the basic advantages and disadvantages of ML.

**1.14.1 Advantages:**

* It can be used for pattern detection.
* It can be used to make predictions about future data.
* It can be used to generate new features from data automatically.
* It can be used to cluster data automatically.
* It can be used to detect outliers in data automatically.

**1.14.2 Disadvantages:**

Some disadvantages include the potential for biased data, overfitting data, and lack of explainability.

**1.15 Machine Learning Algorithms**

There are a variety of machine learning algorithms available and it is very difficult and time consuming to select the most appropriate one for the problem at hand. These algorithms can be grouped in to two categories. Firstly, they can be grouped based on their learning pattern and secondly by their similarity in their function.

Based on their learning style they can be divided into three types:

* + 1. **Supervised Learning Algorithms:** The training data is provided along with the label which guides the training process. The model is trained until the desired level of accuracy is attained with the training data. Examples of such problems are classification and regression. Examples of algorithms used include Logistic Regression, Nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), Neural Networks.
    2. **Unsupervised Learning Algorithms:** Input data is not labeled and does not come with a label. The model is prepared by identifying the patterns present in the input data. Examples of such problems include clustering, dimensionality reduction and association rule learning. List of algorithms used for these type of problems include Apriori algorithm and K-Means and Association Rules .
    3. **Semi-Supervised Learning Algorithms:** The cost to label the data is quite expensive as it requires the knowledge of skilled human experts. The input data is combination of both labeled and unlabelled data. The model makes the predictions by learning the underlying patterns on their own. It is a mix of both classification and clustering problems.

Based on the similarity of function, the algorithms can be grouped into the following:

* + 1. **Regression Algorithms:** Regression is a process that is concerned with identifying the relationship between the target output variables and the input features to make predictions about the new data.  Top six Regression algorithms are: Simple Linear Regression, Lasso Regression, Logistic regression, Multivariate Regression algorithm, Multiple Regression Algorithm.
    2. **Instance-based Algorithms:** These belong to the family of learning that measures new instances of the problem with those in the training data to find out a best match and makes a prediction accordingly. The top instance-based algorithms are: k-Nearest Neighbor, Learning Vector Quantization, Self-Organizing Map, Locally Weighted Learning, and Support Vector Machines.
    3. **Regularization:** Regularization refers to the technique of regularizing the learning process from a particular set of features. It normalizes and moderates. The weights attached to the features are normalized, which prevents in certain features from dominating the prediction process. This technique helps to prevent the problem of overfitting in machine learning. The various regularization algorithms are Ridge Regression, Least Absolute Shrinkage and Selection Operator (LASSO) and Least-Angle Regression (LARS).
    4. **Decision Tree Algorithms:** These methods construct a tree-based model constructed on the decisions made by examining the values of the attributes. Decision trees are used for both classification and regression problems. Some of the well-known decision tree algorithms are: Classification and Regression Tree, C4.5 and C5.0, Conditional Decision Trees, Chi-squared Automatic Interaction Detection and Decision Stump.
    5. **Bayesian Algorithms:** These algorithms apply the Bayes theorem for classification and regression problems. They include Naive Bayes, Gaussian Naive Bayes, Multinomial Naive Bayes, Bayesian Belief Network, Bayesian Network and Averaged One-Dependence Estimators.
    6. **Clustering Algorithms:**Clustering algorithms involve the grouping of data points into clusters. All the data points that are in the same group share similar properties and, data points in different groups have highly dissimilar properties. Clustering is an unsupervised learning approach and is mostly used for statistical data analysis in many fields. Algorithms like k-Means, k-Medians, Expectation Maximisation, Hierarchical Clustering, and Density-Based Spatial Clustering of Applications with Noise fall under this category.
    7. **Association Rule Learning Algorithms:** Association rule learning is a rule-based learning method for identifying the relationships between variables in a very large dataset. Association Rule learning is employed predominantly in market basket analysis. The most popular algorithms are: Apriori algorithm and Eclat algorithm.
    8. **Artificial Neural Network Algorithms:**Artificial neural network algorithms relies find its base from the biological neurons in the human brain. They belong to the class of complex pattern matching and prediction processes in classification and regression problems. Some of the popular artificial neural network algorithms are: Perceptron, Multilayer Perceptrons, Stochastic Gradient Descent, Back-Propagation, , Hopfield Network, and Radial Basis Function Network.
    9. **Deep Learning Algorithms:** These are modernized versions of artificial neural network, that can handle very large and complex databases of labeled data. Deep learning algorithms are tailored to handle text, image, audio and video data. Deep learning uses self-taught learning constructs with many hidden layers, to handle big data and provides more powerful computational resources. The most popular deep learning algorithms are: Some of the popular deep learning ms include Convolutional Neural Network, Recurrent Neural Networks, Deep Boltzmann Machine, Auto-Encoders Deep Belief Networks and Long Short-Term Memory Networks.
    10. **Dimensionality Reduction Algorithms:** Dimensionality Reduction algorithms exploit the intrinsic structure of data in an unsupervised manner to express data using reduced information set. They convert a high dimensional data into a lower dimension which could be used in supervised learning methods like classification and regression. Some of the well known dimensionality reduction algorithms include Principal Component Analysis, Principal Component Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Mixture Discriminant Analysis, Flexible Discriminant Analysis and Sammon Mapping.
    11. **Ensemble Algorithms:** Ensemble methods are models made up of various weaker models that are trained separately and the individual predictions of the models are combined using some method to get the final overall prediction. The quality of the output depends on the method chosen to combine the individual results. Some of the popular methods are: Random Forest, Boosting, Bootstrapped Aggregation, AdaBoost, Stacked Generalization, Gradient Boosting Machines, Gradient Boosted Regression Trees and Weighted Average.

**1.16 Applications of Machine Learning**

These algorithms help in building intelligent systems that can learn from their past experiences and historical data to give accurate results. Many industries are thus applying ML solutions to their business problems, or to create new and better products and services. Healthcare, defense, financial services, marketing, and security services, among others, make use of ML.

**1.16.1 Facial recognition/Image recognition**

The most common application is Facial Recognition, and the simplest example of this application is the iPhone. There are a lot of use-cases of facial recognition, mostly for security purposes like identifying criminals, searching for missing individuals, aid forensic investigations, etc. Intelligent marketing, diagnose diseases, track attendance in schools, are some other uses.

**1.16.2 Automatic Speech Recognition**

Abbreviated as ASR, automatic speech recognition is used to convert speech into digital text. Its applications lie in authenticating users based on their voice and performing tasks based on the human voice inputs. Speech patterns and vocabulary are fed into the system to train the model. Presently ASR systems find a wide variety of applications in the following domains:

* Medical Assistance
* Industrial Robotics
* Forensic and Law enforcement
* Defense & Aviation
* Telecommunications Industry
* Home Automation and Security Access Control
* I.T. and Consumer Electronics

**1.16.3 Financial Services**

Machine learning has many use cases in Financial Services. Machine Learning algorithms prove to be excellent at detecting frauds by monitoring activities of each user and assess that if an attempted activity is typical of that user or not. Financial monitoring to detect money laundering activities is also a critical security use case.

It also helps in making better trading decisions with the help of algorithms that can analyze thousands of data sources simultaneously. Credit scoring and underwriting are some of the other applications. The most common application in our day to day activities is the virtual personal assistants like Siri and Alexa.

**1.16.4 Marketing and Sales**

It is improving lead scoring algorithms by including various parameters such as website visits, emails opened, downloads, and clicks to score each lead. It also helps businesses to improve their dynamic pricing models by using regression techniques to make predictions.

Sentiment Analysis is another essential application to gauge consumer response to a specific product or a marketing initiative. Machine Learning for Computer Vision helps brands identify their products in images and videos online. These brands also use computer vision to measure the mentions that miss out on any relevant text. Chatbots are also becoming more responsive and intelligent.

**1.16.5 Healthcare**

A vital application is in the diagnosis of diseases and ailments, which are otherwise difficult to diagnose. Radiotherapy is also becoming better.

Early-stage drug discovery is another crucial application which involves technologies such as precision medicine and next-generation sequencing. Clinical trials cost a lot of time and money to complete and deliver results. Applying ML based predictive analytics could improve on these factors and give better results.

These technologies are also critical to make outbreak predictions. Scientists around the world are using ML technologies to predict epidemic outbreaks.

**1.16.6 Recommendation Systems**

Many businesses today use recommendation systems to effectively communicate with the users on their site. It can recommend relevant products, movies, web-series, songs, and much more. Most prominent use-cases of recommendation systems are e-commerce sites like Amazon, Flipkart, and many others, along with Spotify, Netflix, and other web-streaming channels.

**1.17 Real-world machine learning use cases**

* **Fraud detection**: Machine learning algorithms can be trained to detect patterns of fraudulent behavior, such as suspicious transactions or fake accounts.
* **Image and speech recognition**: Machine learning algorithms can be used to recognize and classify objects, people, and spoken words in images and audio recordings.
* **Predictive maintenance**: Equipment maintenance can be planned ahead of time to save downtime using machine learning to predict when it is likely to fail.
* **Personalization**: Machine learning can be used to personalize recommendations and advertisements, such as those seen on online shopping websites or streaming services.
* **Healthcare**: Machine learning can be used to predict patient outcomes, identify potential outbreaks of infectious diseases, and assist with diagnosis and treatment planning.
* **Natural language processing**: Machine learning can be used to understand and process human language, enabling applications such as language translation and chatbots.

**1.18 Future of Machine Learning**

Given that machine learning is a constantly developing field that is influenced by numerous factors, it is challenging to forecast its precise future. Machine learning, however, is most likely to continue to be a major force in many fields of science, technology, and society as well as a major contributor to technological advancement. The creation of intelligent assistants, personalized healthcare, and self-driving automobiles are some potential future uses for machine learning. Important global issues like poverty and climate change may be addressed via machine learning.

It is also likely that machine learning will continue to advance and improve, with researchers developing new algorithms and techniques to make machine learning more powerful and effective. One area of active research in this field is the development of artificial general intelligence (AGI), which refers to the development of systems that have the ability to learn and perform a wide range of tasks at a human-like level of intelligence.

**1.19 Introduction To Machine Learning using Python**

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data. In this article, we’ll see basics of Machine Learning, and implementation of a simple machine-learning algorithm using python.

Machine learning is a method of teaching computers to learn from data, without being explicitly programmed. Python is a popular programming language for machine learning because it has a large number of powerful libraries and frameworks that make it easy to implement machine learning algorithms.

To get started with machine learning using Python, you will need to have a basic understanding of Python programming and some knowledge of mathematical concepts such as probability, statistics, and linear algebra.

There are several libraries and frameworks in Python that can be used for machine learning, including:

* scikit-learn: This library provides a wide range of machine learning algorithms, including supervised and unsupervised learning, and it is built on top of other libraries such as NumPy and SciPy.
* PyTorch: An open-source machine learning library for Python, based on Torch library. It provides a seamless integration of computation graph and PyTorch tensors.
* Theano: A numerical computation library for Python that allows you to efficiently define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays.
* Pandas: A library that provides fast and flexible data structures and data analysis tools for the Python programming language.
* **Pygame**: (the library) is a Free and Open Source **python** programming language library for making multimedia applications like games.

**1.20 Jupyter**

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

**1.20.1 Installation**

If so, then you can use a handy tool that comes with Python called pip to install Jupyter Notebook like this:

$ pip install jupyter

The next most popular distribution of Python is [Anaconda](https://www.anaconda.com/). Anaconda has its own installer tool called conda that you could use for installing a third-party package. However, Anaconda comes with many scientific libraries preinstalled, including the

Jupyter Notebook, so you don’t actually need to do anything other than install Anaconda itself.

**CHAPTER 2: TRAINING WORK UNDERTAKEN**

**2.1 TITLE OF PROJECT**:

Snake Game

**2.2 HISTORY OF PROJECT**:

Although, the concept of snake-based games was originated in 1976 and there was resurgence of interest only after a variant was preloaded in Nokia mobiles in 1998. I had also optimized same variant using python programming language. When user presses the Enter key the snake generates a velocity with predefined position in canvas and also a piece of food is also generated at a random position. We can control snake movement according to buttons UP, DOWN, RIGHT, LEFT keys of keyboard. When head of snake collides with the piece of food, the piece of food disappears and size of snake increases by one block and new food is generated at some random position in canvas.

**2.3 DESCRIPTION OF GAME:**

The current snake game is a single player game. The game interface is built in 2D format. The game is displayed in pygme window. Here, it consists of two points where one is movable while the other is fixed. The movable point is snake while the other point is at fixed location known as piece of food. Whenever, the snake gets food the length of snake increases by one. The speed of snake can be altered from the code. The player controls a long, thin creature resembling to snake, which roams around a bordered-walls, searching for food and picking up the food, increases its length by one. The user controls the direction of snake’s head by UP, DOWN, LEFT, RIGHT keys and the snake’s body follows.

A console window is created as soon as game starts. The game will terminate the snake if head collides with any of window’s boundaries or with its own tail or if two keys get presses at same time. Whenever, the game finishes it shows cases Game Over sign and to restart the game again.

**2.4 SYSTEM SPECIFICATIONS**:

• Hardware Requirements:

Processor – AMD Ryzen 5

RAM – 8 GB

Edition – Windows 11

Version – 22 HZ

• Software Requirements:

Operating system – Window 11

Front – End – Jupyter Notebook

Python Language

**2.5 Pre-Requisites of project**

The major topics that we should refresh before we jump straight towards building the Snake game in Python are as follows:

Game development using Pygame & Reinforcement Learning

* Preinstalled **Python 3.0** for building the project on your machine.
* **Pygame:** To build the Snake game in Python we shall be using the open-source library specifically designed for creating video games as it contains inbuilt graphics and sound libraries. This is very beginner-friendly and cross-platform.
* **Time module:** For tracking the number of seconds that have elapsed as compared to the previous time, we make use of the time module.
* **Random module:** With the random module we can generate numbers randomly which will be used as we follow along with the article to build the snake game in Python.
* **Text editor:** We can use either VSCode Python 3 installed on your machine or we can implement the source code with any [online Python compiler](https://www.scaler.com/topics/python/online-python-compiler/) as well or jupyter notebook too.
* **pygame.event.get()**: This helps to trace any action taken by the user on the game window, be it a click or quit. Even if your environment doesn’t require any interaction with the user (as in this case, its the computer doing all the stuff), this snippet is used from time to time in my code else the game window becomes unresponsive. No other major significance
* **clock.tick(1):** This helps in maintaining the frame rate while transitioning on the pygame window. A low value means slow animation & vice-versa. Here 1 as a parameter means 1 frame/second. This can be any number. The bigger the number, the faster the screen transition
* **pygame.display.update():** Every time we draw out anything on the pygame window, this function is required to refresh the screen to reflect the changes done.
* User-defined functions in Python
* Python pygame.Color() function
* pygame .midtop() function
* [for loop in Python](https://www.scaler.com/topics/python/for-loop-in-python/)
* [if statements in Python.](https://www.scaler.com/topics/python-if-else-statement/)
* if- else loop in Python
* [.insert() in Pygame](https://www.scaler.com/topics/insert-in-python/)
* [.fill() in Pygame](https://www.scaler.com/topics/zfill-in-python/)

### **2.6 How are We Going to Build the Snake Game Using Python?**

The algorithm that would help us visualize the whole logical flow of building the Snake game in Python is explained below:

* We start by importing the necessary libraries like pygame, time module and random module. Then we head to defining the snake\_speed as 10 and the width and height of the game window as 960 and 720 respectively.
* The second step includes the initialization of the Pygame and controls the FPS (frame per second) to make the speed of the snake flexible by using the pygame.init() function and implementing the pygame.time.Clock() to control the frames per second.
* Then we start initializing the Snake positions, Food positions, and direction of Snake on the game window. We create the snake by assigning 5 blocks to the snake body and allocate the food for the snake in a random manner within the game window using the random.randrange(start, stop, step) function in Python. We also assign the direction to the snake at the onset of the game.
* Now we head towards, creating UDF (user defined function) for displaying the score earned while playing the snake game in Python where we start by declaring the starting score as 0, followed by creating the UDF function score\_metrics for displaying the score earned inside which we are choosing the font and its related size as well.
* Our fifth step includes building the UDF - end\_gameover() for the scenario when the snake shall hit the wall or eat itself and the Game is OVER! We also give a 5-second wait time by implementing the sleep() function from the [time module in Python](https://www.scaler.com/topics/time-module-in-python/).
* The final step is to create the main function for the entire Snake game in Python to work where we make sure the following scenarios are checked:
  + To validate the keys responsible for the movement of the snake using the for loop in Python and if statements in Python.
  + As we don't want our snake to move in the opposite direction that is RIGHT in our case, instantaneously we shall be implementing the if statements in Python.
  + Snake body growing mechanism - When the snake and its food collide the snake body will grow for which we shall increment the score by 15 and enable the new food to be spanned.
  + If the condition around the game getting 'OVER' once the snake in the snake game in Python hits the wall is validated then the end\_gameover() function is supposed to be called.
  + If the condition around the game getting 'OVER' once the snake in the snake game in Python eats itself is validated then the end\_gameover() function is supposed to be called.
  + Displaying the scores using the score\_metrics function.

**2.7 Steps for Building Snake Game Using Python**

Now as we are equipped with all the libraries, modules, and other requirements needed for the project and we have also refreshed our knowledge of all the topics marked as pre-requisites for building the snake game in Python, let's get our hands dirty and build the Snake game in Python from scratch.

Step 1: Always Import all the Important and Necessary Libraries along with Defining the Game Board

Open the command line interface or any local terminal to install Pygame by using the below code. You also need to install certain important libraries that we shall implement for building the Snake game in Python:

Once our modules are installed we shall start to define the width and height of the window where we shall be playing our game. For displaying the text that will be used in the game, we will define the color in RGB format using the pygame.color() function.

As seen above we are ready with our libraries and module and now we can start building the fun and cool snake game in Python. We also defined the width and height of the text that will be written. Also, we defined the colors for the text that will be visible on the game board by using the pygame.color().

Step 2: Starting to Initialize the Pygame and Control the FPS (frame per second) to Make the Speed of the Snake Flexible

Now we shall start to initialize the Pygame by using the pygame.init() function. After creating the game window as we did in the previous step where we pre-defined the width and height we shall now move towards using the pygame.time.Clock() function to make the speed of the snake flexible enough to adjust to changes in the speed of the snake.

Explanation: In the above code snippet we have initialized the pygame by simply using the pygame.init() function. After which we initialize the window of the game by displaying the caption - Snakes game in Python using the pygame.display.set\_caption() function where we also created a variable named window\_snakegame to set the mode with the width and height of the pre-defined game window. We then control the speed of the snake, by controlling the frame per second using the pygame.time.Clock() function.

Step 3: Initializing the Snake Positions, Food Positions, and Direction of Snake on the Game Window

With this step, we shall be initializing the snake's position in the game window. After that is done, we also need to initialize the food's position random manner in the defined width and height of the game window. Also, when the game starts we need the snake to move in a particular direction so that every time the user runs the program of the snake game in Python, the snake moves in that particular set direction. For this code base, we are setting the direction of the snake as 'LEFT'.

Explanation: As viewed from the snippet of the code, we have first defined the snake's position on the game window by the variable - position\_snake. Then we created the snake body length by allocating 5 blocks from the game window as can be seen from the code for the variable - body\_snake. Now, for placing the food in a random manner inside the game window, we made use of the random module function random.randrange(start, stop, step) from which we select a random number within the specified range. As we want to set the direction of the snake to be in LEFT so that every time a user starts the game the snake starts to travel from the LEFT side of the game window.

Step 4: Creating UDF for Displaying Score Earned While Playing the Snake Game in Python.

In the fourth step, we shall be creating a UDF(user-defined function) that will build a font object where we shall define the font color with an appropriate size allocated for the score variable. Then we shall make use of the render function in Python to build the background surface which will change as the score updates. As the text gets refreshed, we need to create a rectangular object referencing the text surface object by implementing the .get\_rect() function. Using the blit() function we will display the score that is earned while playing the snake game in Python. For the screen.blit(background,(x,y)) function usually takes two arguments where (x,y) is the position inside the window where we want the top left of the surface to be. The function tells to take the background surface and draw it onto the screen by positioning it at (x,y).

Explanation: The code snippet illustrates the UDF score\_metrics created for the score and its related metrics that are being earned while playing the snake game in Python. The score earned could be displayed by calling the score\_metrics UDF and the updated score is displayed.

Step 5: Building the UDF for a Scenario When the Snake shall Hit the Wall or Eat itself and the Game is OVER!

We shall be creating a UDF for defining the condition around the game getting 'OVER' once the snake in the snake game in Python either hits the wall or eats itself. For doing so, we shall start by first creating a font object which will display the scores earned so far. To render scores, we create a text surface for the text to be drawn, for which we again create a rectangular object for the text surface object.

As we want the message of "GAME OVER" to appear in the middle of the game window of the snake game in Python, we set the position of the text surface object in the middle. Then as we need to show the text on screen we use the blit() function. As the game goes on, the score updates which we shall display by updating the surface by implementing the flip() function. Before we can close the game window, we give a 5-second wait time by implementing the sleep() function and eventually quit the game by using pygame.quit() function.

Explanation:With the code written above, we created the UDF end\_gameover that covers the scenario when the snake shall hit the wall or eat itself and the Game is OVER! We have declared variable and implemented functions like variable font\_displayscore for creating a font object which will display the scores earned so far, variable surface\_gameover for text surface object on which the text for scoring and all will be written, variable rect\_gameover to create a rectangular object for the text surface object, window\_snakegame.blit() function to show the text on screen we using the blit() function and pygame.display.flip() function for updating the score by updating the surface using flip() function. Eventually, we want to give a 5-second wait time by implementing the sleep() function and finally quit the game by using pygame.quit() function.

Step 6: Creating the Main Function for the entire Snake Game in Python to Work.

Creating the main function will allow the snake game in Python to do the following listed points:

To validate the keys responsible for the movement of the snake using the for loop in Python and if statements in Python.

As we don't want our snake to move in the opposite direction that is RIGHT in our case, instantaneously we shall be implementing the if statements in Python to resolve this.

We want to make sure that when the snake and its food collide the snake body will grow for which we shall increment the score by '15' and enable the new food to be spanned. We shall do this by using the if- else loop in Python, .insert() in Pygame , .fill() in Pygame and random.randrange() in Python.

Checking if the condition around the game getting 'OVER' once the snake in the snake game in Python either hits the wall or eats itself is validated then the end\_gameover() function is supposed to be called.

Displaying the scores using the score\_metrics function.

Explanation: As can be remarked from the code written above, we have created the main function that will help us to play the snake game in Python keeping in mind the rules and objective of the game. We validated the keys responsible for the movement of the snake using the for loop in Python and if statements in Python. We also checked if our snake moves in the opposite direction that is, RIGHT in our case, instantaneously we shall be implementing the if statements in Python to resolve this. When the snake and its food collide the snake body will grow for which we shall increment the score by 15 and enable the new food to be spanned. We shall do this by using the if- else loop in Python, .insert() in Pygame , .fill() in Pygame and random.randrange() in Python. If the condition around the game getting 'OVER' once the snake in the snake game in Python either hits the wall or eats itself is validated then the end\_gameover() function is supposed to be called. Lastly, we are displaying the scores using the score\_metrics function.

It's Gameplay Time!

We shall now combine all the code snippets that we just created and see how the snake game in Python works.

Output:

snake game

Extra Features That Can Be Added To The Game

Hurray! Now that you have to build your snake game in Python by following along with the article, we can now start to challenge the technicality and play around with Python by adding a few new features in the snake game in Python. For doing so, we can use the pre-installed Turtle library to implement the snake game in Python. By providing the user with a virtual canvas, the turtle library allows its users to print patterns and images.

You can try experimenting with the turtle library in Python to explore more features in the snake game in Python.

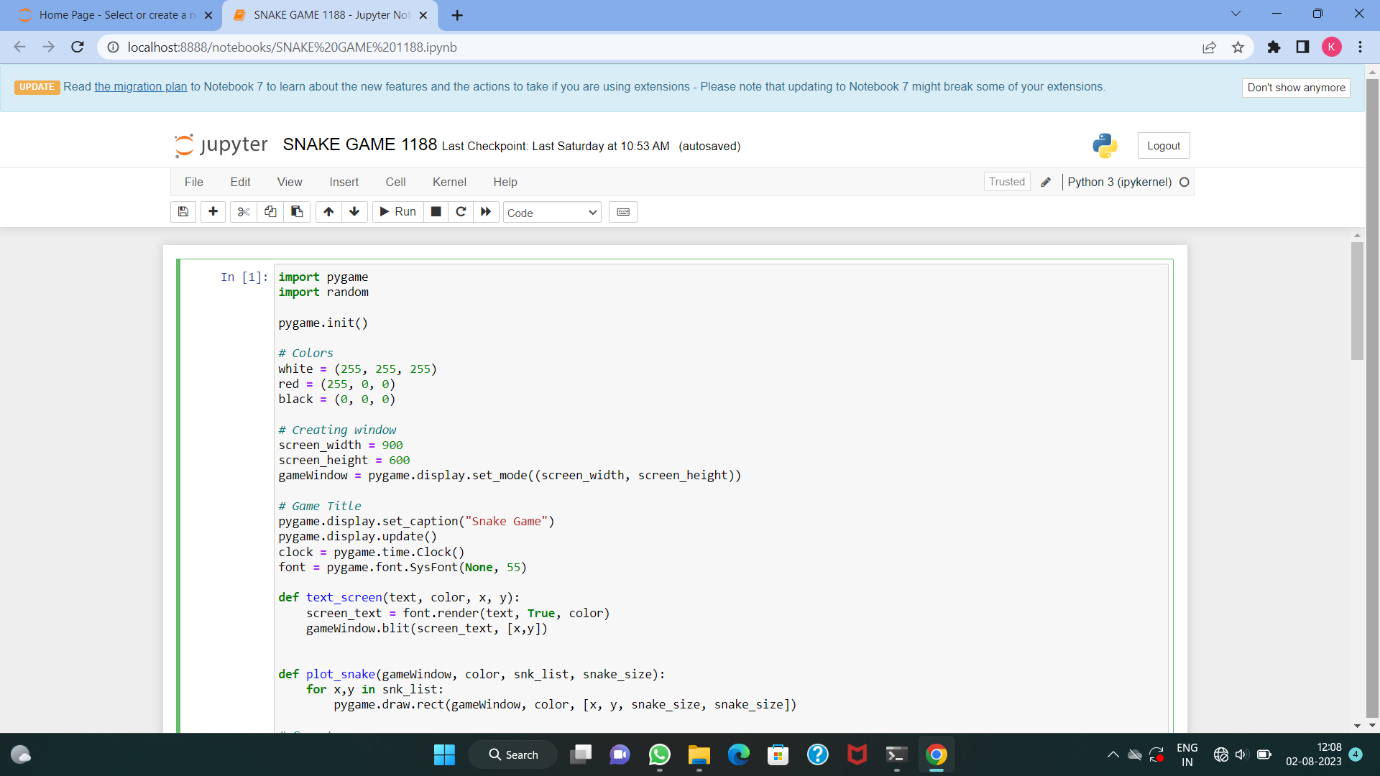
Conclusion

The Snake game in Python is controlled with the four direction buttons on our keyboard concerning the direction in which the snake runs with the clear goal of eating maximum food and eventually scoring maximum points until the snake strikes the wall or itself.

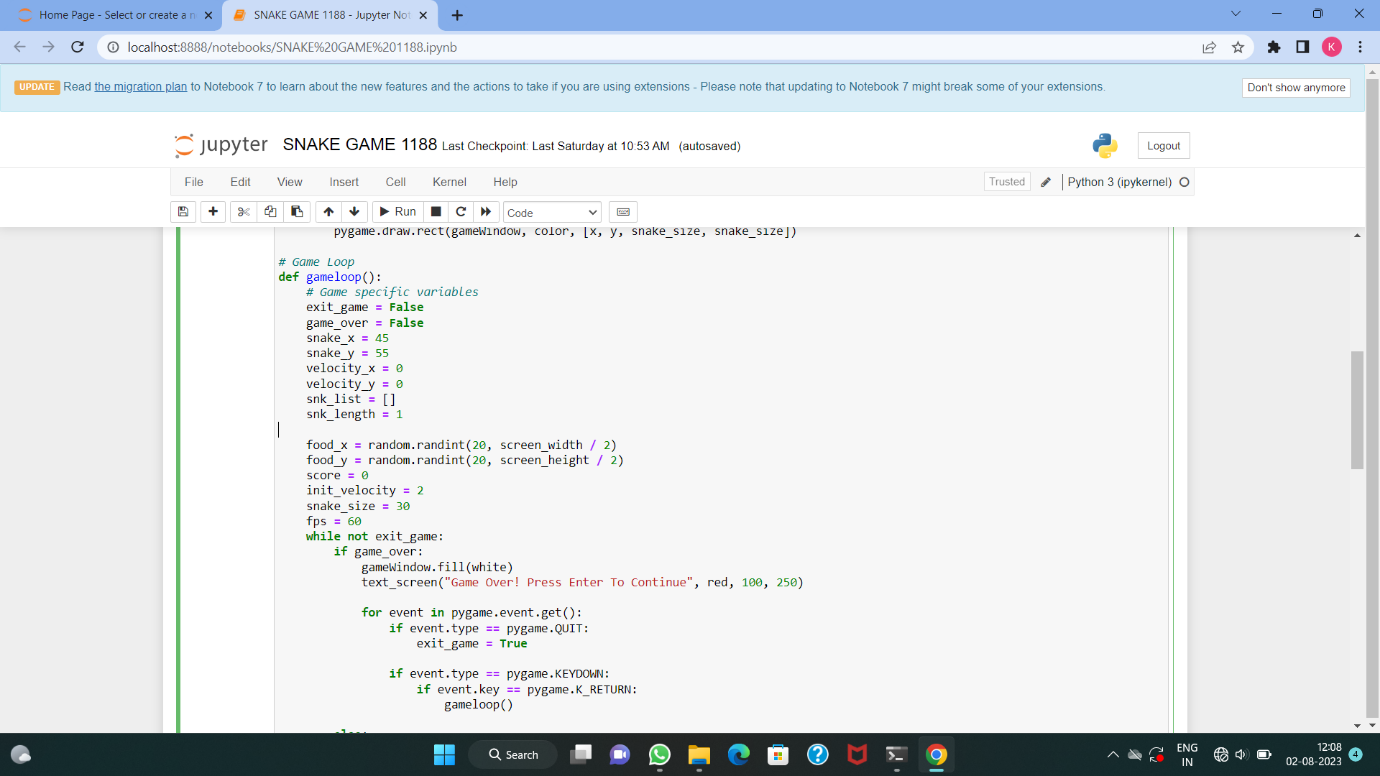
It's recommended to refresh the pre-requisites section before you start to build the snake game in Python.

While building any game all the scenario that may arise while playing the game needs to be handled and hence we provided the algorithm before jumping into the coding part where the foundation of the logical flow of the snake game in Python was laid out.

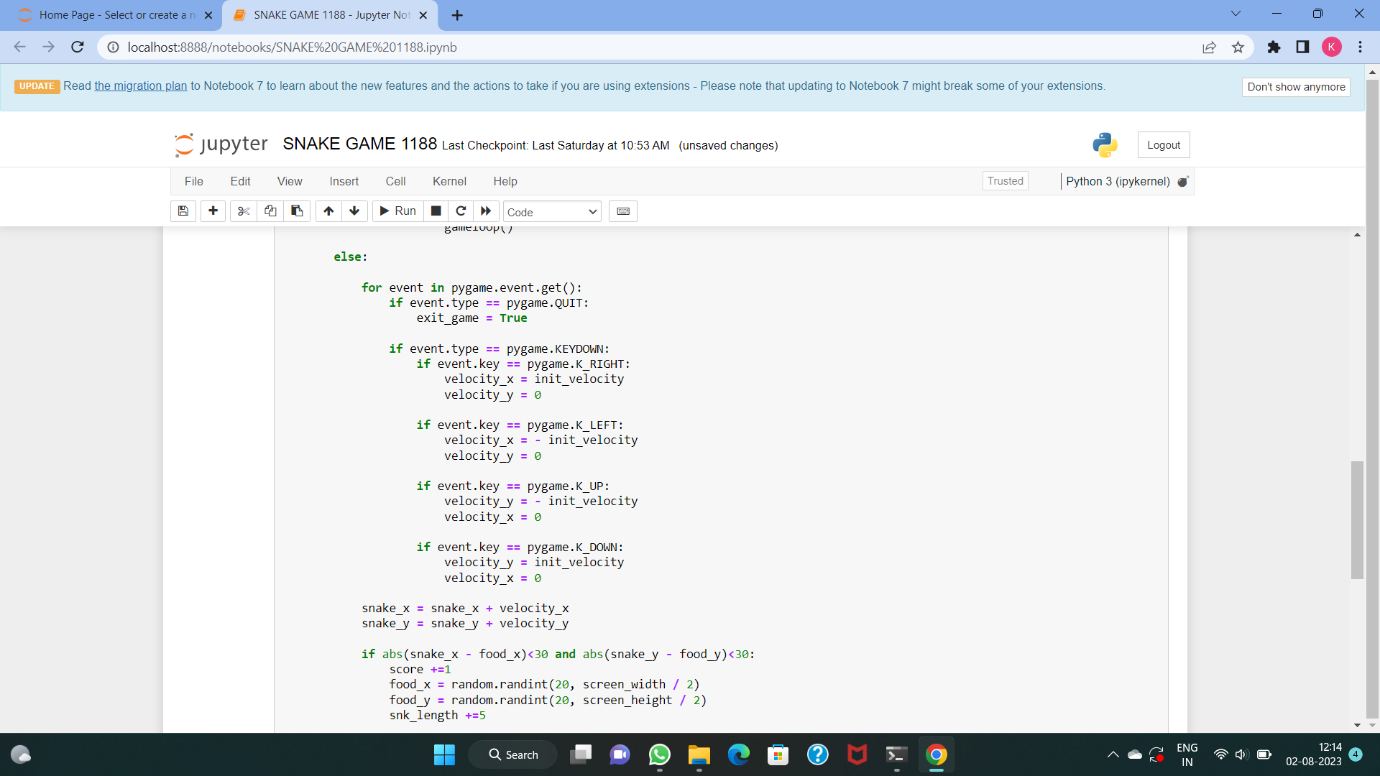
**2.8 Screenshots of Project:**

****

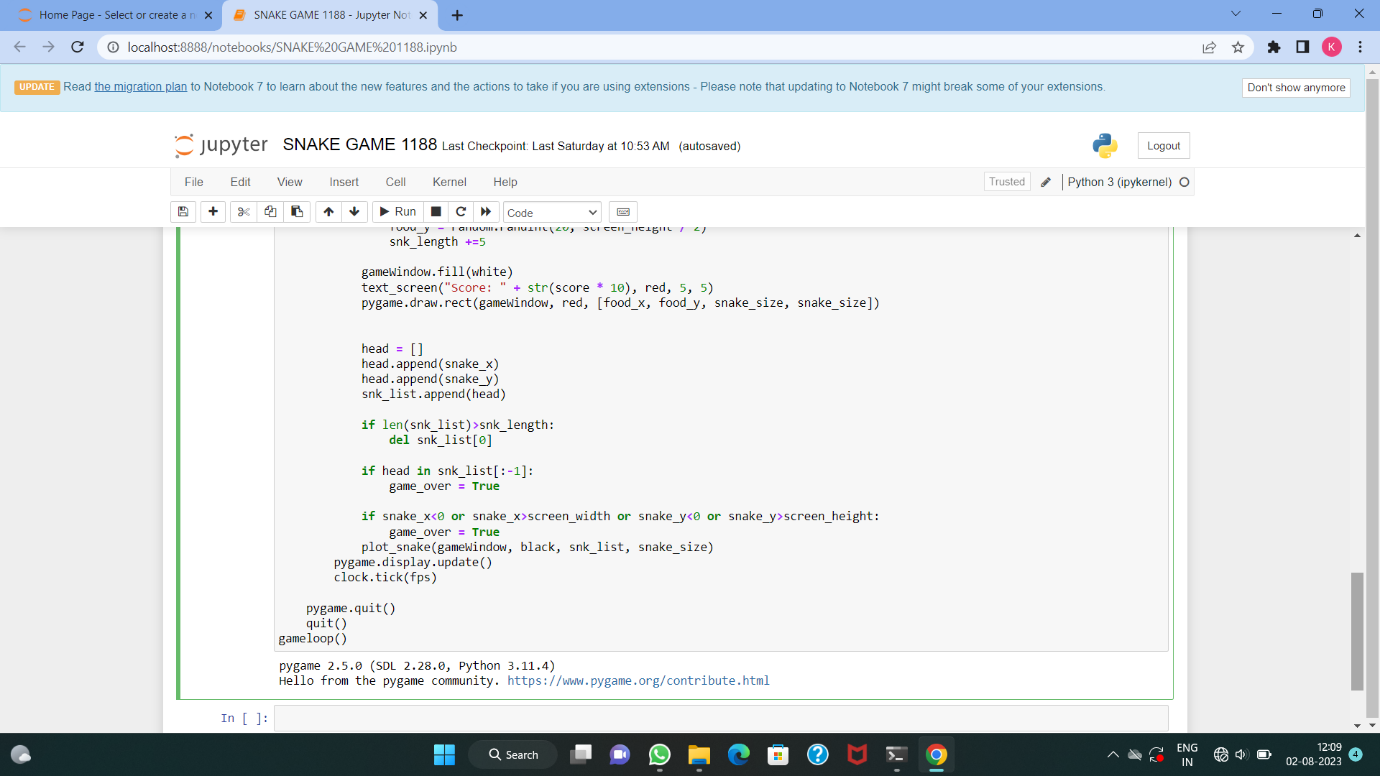
*Figure2.1*

**

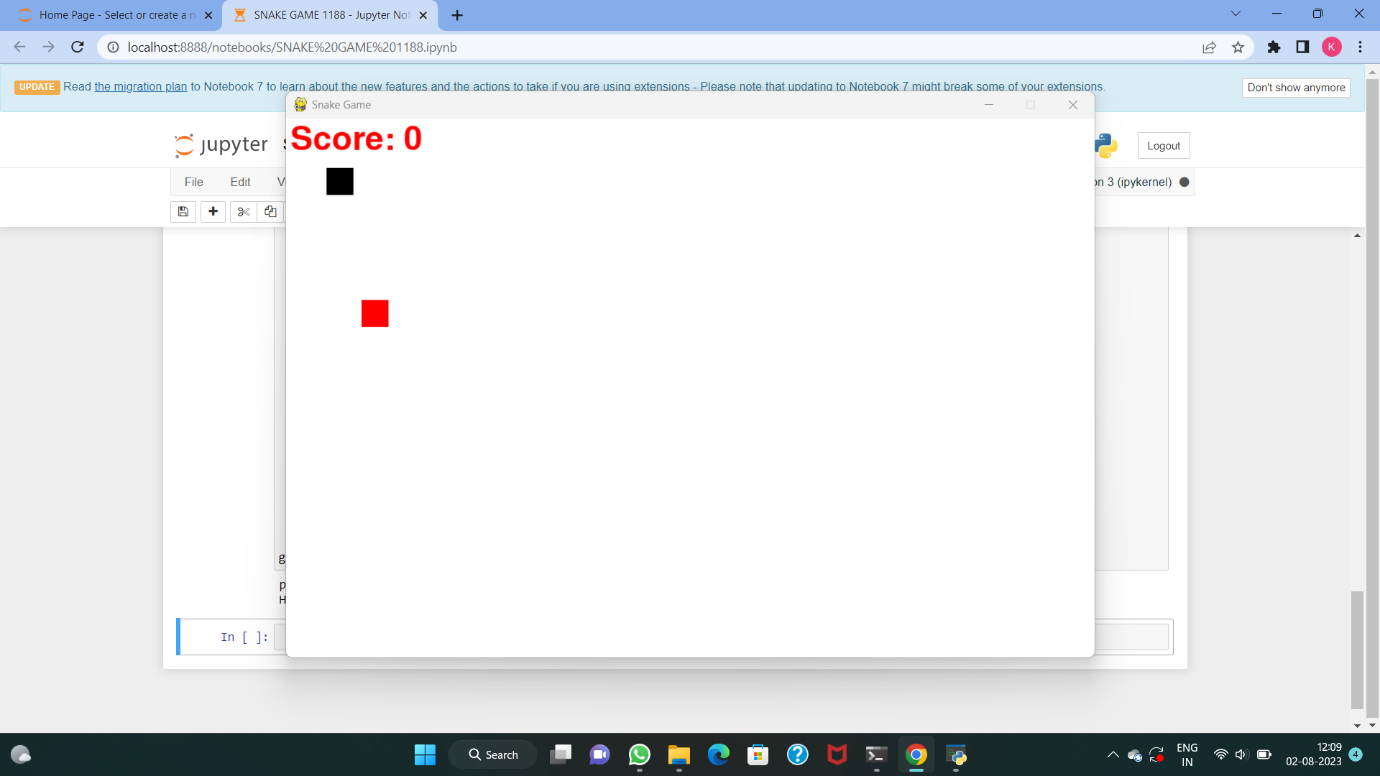
*Figure2.2*

****

*Figure2.3*

****

*Figure2.4*

****

*Figure2.5*

****

*Figure2.6*

****

*Figure2.7*

****

*Figure2.8*

**CHAPTER 3: RESULT AND DISCUSSION**

**3.1 RESULT:**

Snake Game project is written in Python. The project file contains Assets, python scripts (main.py, food.py, block.py, world.py) with text files. The gameplay Graphics is simple, good enough and the controls are simple for the users. Talking about the gameplay, it’s a common game for all. All the playing methods are too simple like the original all you have to do is just move around in order to eat things. Here, the user has to control the snake using A, S, D, W keys and the snake expands after eating things. After a certain score, the borders will appear in the game. The player loses whenever the snake runs into a border or itself/its own tail. A simple GUI is provided for the easy gameplay. The gameplay design is so simple that user won’t find it difficult to use and navigate. In order to run the project, you must have installed [Python](https://www.python.org/downloads/release/python-365/), Pygame on your PC. This is a simple GUI Based game written for the beginners.

**3.2 DISCUSSION:**

I had made my project on Snake Game. Basically, my project was for beginners who wants to get knowledge of basic game play in pygame module. Another major initiative of this project was to revive our childhood. Once it our lives we all had played this game either in our daddy’s mobile phone or computers or our all time favourite Nokia Keypad mobile phone. I think we all had once played in Nokia phones.

Today, in this busy chaotic schedule where we had lost our once personality in this adulthood Lets get back those Golden days of childhood full of enjoyment, never ending chatterings, Childhood is the most fun and memorable time in anyone’s life. It’s the first stage of life which we enjoy in whatever way we like. Besides, this is the time that shapes up the future. The parents love and care for their children and the children to the same too. Moreover, it’s the golden period of life in which we can teach children everything. The [memories of childhood](https://www.toppr.com/guides/essays/childrens-day-essay/) ultimately become the life long memory which always brings a smile on our faces. Only the grownups know the real value of childhood because the children do not understand these things. Moreover, Children’s have no worries, no stress, and they are free from the filth of worldly life. Also, when an individual collects memories of his/her childhood they give a delighted feeling. Snakeis a classic game that requires players to assess their surroundings and find the quickest or safest route to a point. This is an excellent opportunity to learn about spatial awareness and plan ahead to your next move. The classic game is infamous for using your own success against you when you become so long that you get in your own way.

**CHAPTER4: CONCLUSIONS AND FUTURE SCOPE**

**4.1 Conclusion:**

The Snake game in Python is controlled with the four direction buttons on our keyboard concerning the direction in which the snake runs with the clear goal of eating maximum food and eventually scoring maximum points until the snake strikes the wall or itself. It's recommended to refresh the pre-requisites section before you start to build the snake game in Python. While building any game all the scenario that may arise while playing the game needs to be handled and hence we provided the algorithm before jumping into the coding part where the foundation of the logical flow of the snake game in Python was laid out.

**4.2 Future Scope:**

Our project will be able to implement in future after making some changes and modifications as we make our project at a low level.

* It can be made with good Graphics
* We can add more options like Top scores and player’s profile
* We can add multiplayer option
* Envision multimedia-based expertise and cross-media growth strategies
* Visualize quality intellectual capital
* Engage worldwide methodologies with web-enabled technologies

**4.3 References:**

* Javatpoint
* Geeksforgeeks
* Python.com
* Pygame.com