# Lecture 1
# Introduction to Machine Learning Algorithms

ASTANA IT UNIVERSITY

Sarinova Assiya

Associated professor, PhD in CSaSI

2022

# Course organization

- Course activities
  - Attend **2 hours** lectures per week
  - The total point of Attendance is **70-80%** or more to pass every separate exam (Midterm, Endterm, and Final exams).
- Lecture notes available at least one day prior to lecture
  - Work on the workshop questions
- Will be discussed during the following week's workshop which follows immediately after the 2-hour lecture
  - Work on the home exam
- Topic for the assignment can be freely chosen.
- Not just about facts, you also need to
  - understand concepts
  - apply those concepts
  - think about implications
  - understand limitations

# Lecturer

- **Associated Prof. Sarinova Assiya**
- **Education**
  - Specailitet CSaPDaM (2008)
  - Msc Informatiks (2011)
  - Candidate of Technical Sciences (2019, specialty: mathematical and software support o machines, complexes and computer networks, Tomsk State University)
  - PhD (2020, specialty: Computer Engineering and Software, nostrification of the Republic of Kazakhstan)
- **Work**

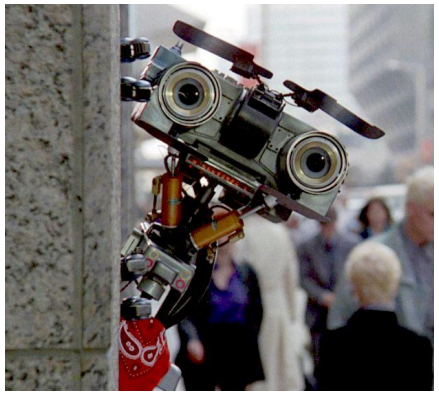
| | |
|---|---|
| 2009-2014 | assistant teacher, St. Rev. Department "ASOIiU", InEU |
| 2014-2019 | Software Engineer, senior lecturer of the Department "VTiP" of S. Toraigyrov PSU. |
| 2019-2020 | senior lecturer. Department of "Electrical Engineering and Automation" of NAO "Toraigyrov University". |
| 2020-2021 | Associate Professor (Associate Professor) "Electrical Engineering and Automation" of NAO "Toraigyrov University". |
| 2021-2022 | Senior lecturer of the Department of Electrical Equipment Operation of the Kazakh Agrotechnical University named after Saken Seifullin |
| 2021-2022 | Head of the Department of Information Systems and Technologies |
| 2022- Until now | Associated Proffesor of Department of Intelligent systems and cybersecurity |

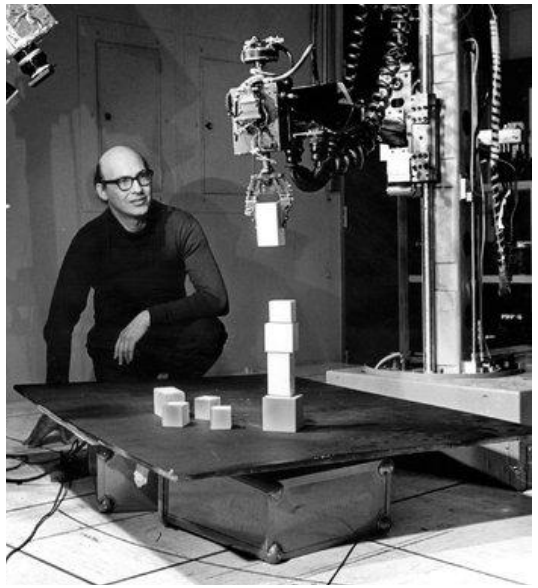
# Artificial Intelligence

## Worldwide A.I. investment to top $200bn by 2025

*KPMG.* July 31, 2018

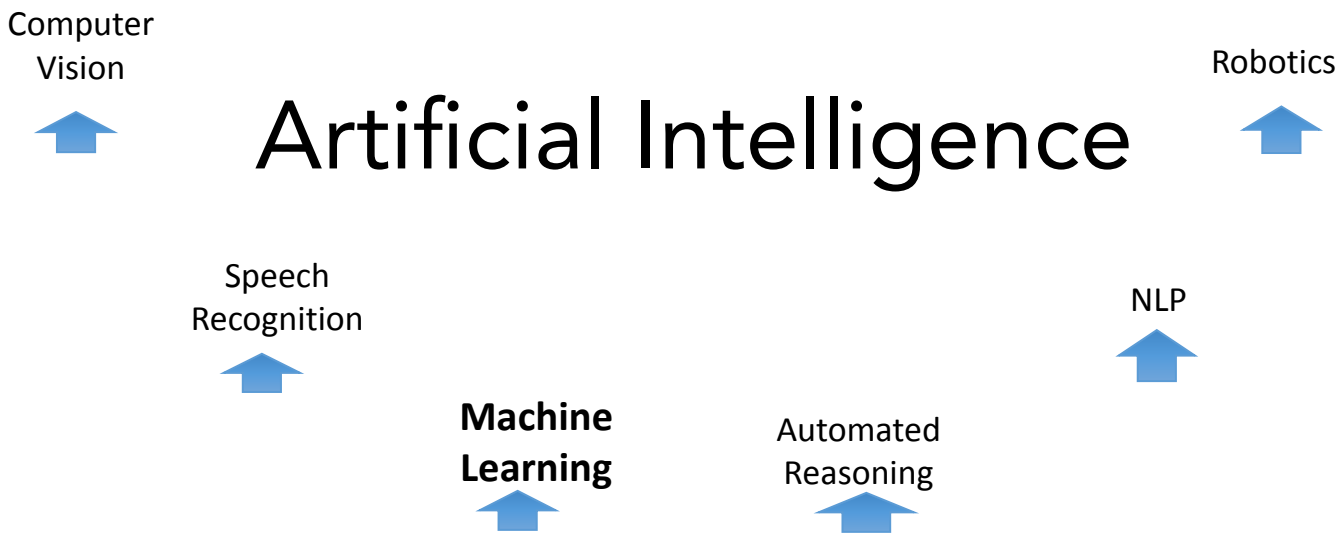
*"We view AI as an ecosystem that unlocks value by enhancing, accelerating, and automating decisions that drive growth and profitability."*



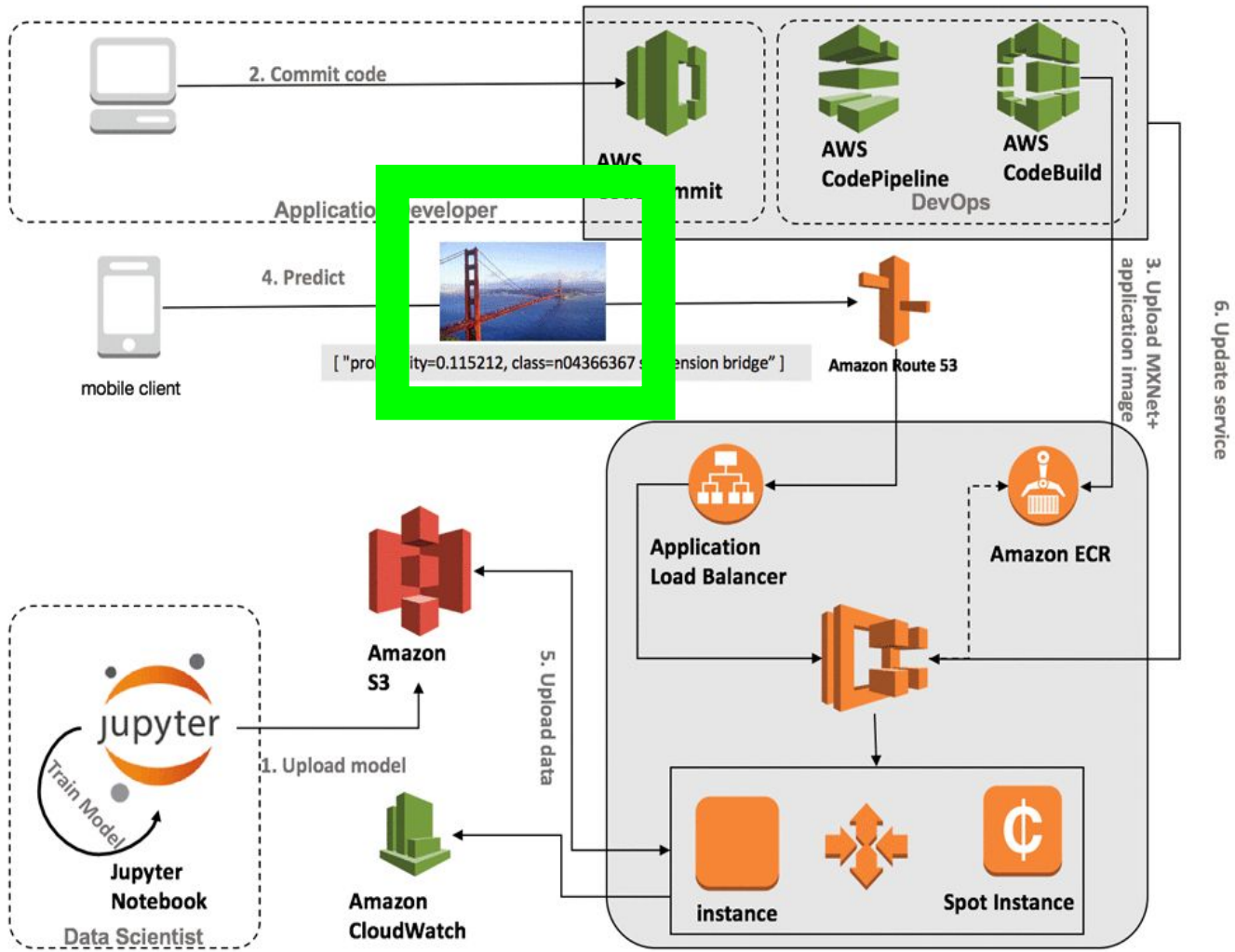



Google
self-driving car


amazon




Arthur Samuel (1959): Machine Learning is the field of study that gives the computer the ability to learn without being explicitly programmed.


2. Commit code
AWS
mmit
AWS
CodePipeline
AWS
CodeBuild
DevOps
Applicatio eveloper
4. Predict
mobile client
[ "pro ity=0.115212, class=n04366367 ension bridge" ]
Amazon Route 53
3. Upload MXNet+
application image
6. Update service
Application
Load Balancer
Amazon ECR
Amazon
S3
5. Upload data
jupyter
Train Model
1. Upload model
Jupyter
Notebook
Data Scientist
Amazon
CloudWatch
instance
Spot Instance

# About Machine learning

- Machine learning is about extracting knowledge from data. It is a scientific field located at the intersection of statistics, artificial intelligence and computer science and is also known as predictive analytics or statistical learning. In recent years, the use of machine learning methods in everyday life has become commonplace.
- Many modern websites and devices use machine learning algorithms, starting with automatic recommendations for watching movies, ordering food or buying groceries, and ending with personalized online radio broadcasts and recognizing friends in photos. When you see a complex site like Facebook, Amazon or Netflix, it is very likely that each section of the site contains several machine learning models.

# About Machine learning

- Machine learning is a multidisciplinary field created at the intersection of, and with synergy between, computer science, statistics, neurobiology, and control theory.
- It has played a key role in various fields and has radically changed the vision of programming software. For humans, and more generally, for every living being, learning is a form of adaptation of a system to its environment through experience.
- This adaptation process must lead to improvement without human intervention. To achieve this goal, the system must be able to learn, which means that it must be able to extract useful information on a given problem by examining a series of examples associated with it.

# Why do I need to use machine learning

- At the dawn of the emergence of "intelligent" applications, many systems used strict "if" and "else" rules to process data or correct information entered by the user. Think of the spam filter, whose job is to move the corresponding incoming emails to the Spam folder.
- You can create a blacklist of words that will identify the email as spam. This is an example of using a system of expert rules to develop an "intelligent" application.The development of decision-making rules in manual mode is acceptable in some tasks, especially in those where people clearly understand the modeling process.

# However, the use of rigid decision rules has two main drawbacks:

- The logic needed to make a decision relates exclusively to one specific area and task. Even an insignificant change in the task may entail are write of the entire system.
- Developing rules requires a deep understanding of the decision-making process.

# Example

- One example where this rigid approach will fail is face recognition in images.
- Today, every smartphone can recognize a face in an image. However, facial recognition has been an unsolved problem, at least until2001. The main problem is that the way in which a computer "perceives" pixels forming an image on a computer is very different from a human oneperception of the face.
- This difference in principle does not allow a person to formulate a suitable set of rules describing a face from the point of view of a digital image. However, thanks to machine learning, simply presenting a large number of images with faces will be enough for the algorithm to determine which features are necessary forface identification.

# Tasks that can be solved using machine learning

- The most successful machine learning algorithms are those that automate decision–making processes by generalizing well-known examples. In these methods, known as supervised learning or supervised learning, the user provides the algorithm with an object-response pair, and the algorithm finds a way to get an answer by object.
- In particular, the algorithm is able to give an answer for an object that it has never seen before, without any human help. If we go back to the example of spam classification using machine learning, the user presents the algorithm with a large number of emails (objects) along with information about whether the email is spam or not (responses). For a new email, the algorithm calculates the probability with which this email can be attributed to spam.

# Machine learning algorithms

- Machine learning algorithms that learn from pairs of object response are called learning algorithms with a teacher, since the "teacher" shows the algorithm the answer in each observation, according to which the training takes place.
- Despite the fact that creating a set with objects and answers is often a laborious process carried out manually, learning algorithms with a teacher are interpretable and the quality of their work is easy to measure. If your task can be formulated as a learning task with a teacher, and you can create a dataset that includes answers, machine learning will probably solve your problem.

# Examples of machine learning tasks with a teacher:

- **Determining the postal code** by the handwritten numbers on the envelope Here the object will be a scanned image of the handwriting, and the answer will be the actual numbers of the postal code. To create a dataset for building a machine learning model, you need to collect a large number of envelopes. Then you can read the zip codes yourself and save the numbers as answers.
- **Determination of tumor goodness** based on medical images Here the object will be the image, and the answer is the diagnosis of whether the tumor is benign or not. To create a dataset for building a model, you need a database of medical images. In addition, an expert opinion is needed, so the doctor should review all the images and decide which tumors are benign and which are not. In addition to image analysis, additional diagnostics may be needed to determinethe benign nature of the tumor.
- **Detection of fraudulent activity** in credit card transactions. Here the object is a record of a credit card transaction, and the answer is information about whether the transaction is fraudulent or not.Suppose you are a credit card issuing institution, data collection involves saving all transactions and recording customer messages about fraudulent transactions.

# Discuss examples

- Having given these examples, it is interesting to note that although the objects and answers look quite simple, the data collection process for these three tasks is significantly different.
- Despite the fact that reading envelopes isa time-consuming activity, this process is simple and cheap.
- Obtaining medical images and conducting diagnostics requires not only expensive equipment, but also rare, highly paid expert knowledge, not to mention ethical issues and privacy issues. In the example of detecting credit card fraud, data collection is much easier. Your customers will provide you with answers themselves, reporting fraud.
- All you have to do to get objects and responses related to fraudulent activity is to wait.

# Unsupervised learning algorithms

- **Unsupervised learning algorithms** (unsupervised algorithms) are another type of algorithms. In unsupervised learning algorithms, only objects are known, and there are no answers. Although there are many successful applications of these methods, they tend to be more difficult to interpret and evaluate.
- **Examples of machine learning tasks without a teacher:**
- Identifying topics in a set of posts If you have a large collection of text data, you can aggregate them and find common topics. You have no preliminary information about what topics are covered there and how many of them. So there are no known answers.

# Examples of machine learning tasks without a teacher:

- **Segmenting customers** into groups with similar preferences Having a set of customer records, you can identify groups of customers with similar preferences. For a shopping site, suchgroups can be "parents", "bookies" or "gamers".Since you don't know in advance about the existence of these groups and their number, you have no answers.
- **Detecting patterns** of abnormal behavior on a website In order to identify abuses or errors, it is often useful to find patterns of behavior that differ from the norm. The patterns of abnormal behavior may be different, and you may not havethere will be no reported cases of abnormal behavior.Since in this example you are only observing traffic, and you do not know what constitutes normal and abnormal behavior, we are talking about the task of teaching without a teacher.

# Machine learning tasks without a teacher:

- When solving machine learning tasks with and without a teacher, it is important to present your input data in a format that is understandable to a computer.
- Often the data is presented in the form of a table. Every data point you want to explore (every email, every customer, every transaction) is a row, and every property that describes that data point (say, customer age, amount, or transaction location) is a column. You can describe users by age, gender, account creation date and frequency of purchases in your online store. You can describe the image of the tumor using grayscale for each pixel or using the size, shape and color of the tumor.

# Discuss examples

- In machine learning, each object or row is called a sample or a data point, and the columns-properties that describe these examples are called characteristics or features.
- Later we will focus in more detail on the topic of data preparation, which is called feature extraction or feature engineering. However, you should keep in mind that no machine learning algorithm will be able to make a prediction based on data that does not contain any useful information.
- For example, if the only sign of a patient is his last name, the algorithm will not be able to predict his gender. This information is simply not in the data. If you add one more sign – the name of the patient, then things will already be better, because often, knowing the name of a person, you can judge his gender.

# Science with Python

- The amount of digital data that exists is growing at a rapid rate, doubling every two years, and changing the way we live. It is estimated that by 2020, about 1.7MB of new data will be created every second for every human being on the planet. This means we need to have the technical tools, algorithms, and models to clean, process, and understand the available data in its different forms for decision-making purposes.
- *Data science* is the field that comprises everything related to cleaning, preparing, and analyzing unstructured, semistructured, and structured data. This field of science uses a combination of statistics, mathematics, programming, problem-solving, and data capture to extract insights and information from data.

# The Stages of Data Science

- Figure 1-1 shows different stages in the field of data science. Data scientists
- use programming tools such as Python, R, SAS, Java, Perl, and C/C++
- to extract knowledge from prepared data. To extract this information,
- they employ various fit-to-purpose models based on machine leaning
- algorithms, statistics, and mathematical methods.


***Figure 1-1.*** *Data science project stages*

# Why Python?

- Python is  a dynamic and general-purpose programming language that is used in various fields. Python is used for everything from throwaway scripts to large, scalable web servers that provide uninterrupted service 24/7.
- It is used for web programming, and application testing. It is used by scientists writing
- applications for the world's fastest supercomputers and by children first learning to program. It was initially developed in the early 1990s by Guido van Rossum and is now controlled by the not-for-profit Python Software Foundation, sponsored by Microsoft, Google, and others.
- The first-ever version of Python was introduced in 1991. Python is now at version 3.*x*, which was released in February 2011 after a long period of testing. Many of its major features have also been backported to the backward-compatible Python 2.6, 2.7, and 3.6. GUI and database programming, client- and server-side

# Basic Features of Python

## Python provides numerous features; the following are some of these important features:

- • ***Easy to learn and use*****:** Python uses an elegant syntax, making the programs easy to read. It is developer-friendly and is a high-level programming language.
- • ***Expressive***: The Python language is expressive, which means it is more understandable and readable than other languages.
- • ***Interpreted***: Python is an interpreted language. In other words, the interpreter executes the code line by line. This makes debugging easy and thus suitable for beginners.
- • ***Cross-platform*****:** Python can run equally well on different platforms such as Windows, Linux, Unix, Macintosh, and so on. So, Python is a portable language.
- • ***Free and open source*****:** The Python language is freely available at www.python.org. The source code is also available.

# Basic Features of Python

- *Object-oriented*: Python is an object-oriented language with concepts of classes and objects.
- • *Extensible*: It is easily extended by adding new modules implemented in a compiled language such as C or C++, which can be used to compile the code.
- • *Large standard library*: It comes with a large standard library that supports many common programming tasks such as connecting to web servers, searching text with regular expressions, and reading and modifying files.
- • *GUI programming support*: Graphical user interfaces can be developed using Python.
- • *Integrated*: It can be easily integrated with languages such as C, C++, Java, and more.

# Portable Python Editors (No Installation Required)

- These editors require no installation:
- ***Azure Jupyter Notebooks***: The open source Jupyter Notebooks was developed by Microsoft as an analytic playground for analytics and machine learning.
- ***Python(x,y)***: Python(x,y) is a free scientific and engineering development application for numerical computations, data analysis, and data visualization based on the Python programming language, **Qt graphical user interfaces**, and Spyder interactive scientific development environment.
- ***WinPython***: This is a free Python distribution for the Windows platform; it includes prebuilt packages for ScientificPython.
- ***Anaconda***: This is a completely free enterpriseready Python distribution for large-scale data processing, predictive analytics, and scientific computing.

# Tabular Data and Data Formats

- Data is available in different forms. It can be unstructured data, semistructured data, or structured data.
- Python provides different structures to maintain data and to manipulate it such as variables, lists, dictionaries, tuples, series, panels, and data frames. Tabular data can be easily represented in Python using lists of tuples representing the records of the data set in a data frame structure.
- Though easy to create, these kinds of representations typically do not enable important tabular data manipulations, such as efficient column selection, matrix mathematics, or spreadsheet-style operations. Tabular is a package of Python modules for working with tabular data. Its main object is the tabarray class, which is a data structure for holding and manipulating tabular data. You can put data into a tabarray object for more flexible and powerful data processing.
- The Pandas library also provides rich data structures and functions designed to make working with structured data fast, easy, and expressive. In addition, it provides a powerful and productive data analysis environment.
- A Pandas data frame can be created using the following constructor:

*pandas.DataFrame( data, index, columns, dtype, copy)*

# Pandas data frame

- A Pandas data frame can be created using various input forms such as the following:
- List
- Dictionary
- Series
- Numpy ndarrays

Another data frame

# Python Pandas Data Science Library

- Pandas is an open source Python library providing high-performance data manipulation and analysis tools via its powerful data structures. The name Pandas is derived from “panel data,” an econometrics term from multidimensional data. The following are the key features of the Pandas library:
- Provides a mechanism to load data objects from different formats
- Creates efficient data frame objects with default and customized indexing
- Reshapes and pivots date sets
- Provides efficient mechanisms to handle missing data
- Merges, groups by, aggregates, and transforms data
- Manipulates large data sets by implementing various functionalities such as slicing, indexing, subsetting, deletion, and insertion
- Provides efficient time series functionality

# Technical requirements

- We will use various Python packages, such as NumPy, SciPy, scikit-learn, and Matplotlib, during the course of this book to build various things. If you use Windows, it is recommended that you use a SciPy-stack-compatible version of Python. You can check the list of compatible versions at http:/ / www. scipy. org/ install. html. These distributions come with all the necessary packages already installed. If you use MacOS X or Ubuntu, installing these packages is fairly straightforward. Here are some useful links for installation and documentation:
- **NumPy: https:/ / www. numpy. org/ devdocs/ user/ install. html.**
- **SciPy: http:/ / www. scipy. org/ install. html.**
- **Scikit-learn: https:/ / scikit- learn. org/ stable/ install. html.**
- **Matplotlib: https:/ / matplotlib. org/ users/ installing. html.**

# A Pandas Series

- A *series* is a one-dimensional labeled array capable of holding data of any type (integer, string, float, Python objects, etc.). Listing 1 shows how to create a series using the Pandas library.

```
In [1]: #Create series from array using pandas and numpy
        import pandas as pd
        import numpy as np
        data = np.array([90,75,50,66])
        s = pd.Series(data,index=['A','B','C','D'])
        print (s)
```

```
A    90
B    75
C    50
D    66
dtype: int32
```

```
In [5]: #Create series from dictionary using pandas
        import pandas as pd
        import numpy as np
        data = {'Ahmed' : 92, 'Ali' : 55, 'Omar' : 83}
        s = pd.Series(data,index=['Ali','Ahmed','Omar'])
        print (s)
```

```
Ali      55
Ahmed    92
Omar     83
```

# A Pandas Data Frame

- A *data frame* is a two-dimensional data structure. In other words, data is aligned in a tabular fashion in rows and columns. In the following table, you have two columns and three rows of data. Listing 2 shows how to create a data frame using the Pandas library.

```
In [10]: # Creating a Data Frame Using the Pandas Library
         import pandas as pd
         data = [['Ahmed',35],['Ali',17],['Omar',25]]
         DataFrame1 = pd.DataFrame(data,columns=['Name','Age'])
         print (DataFrame1)
```

```
    Name  Age
0  Ahmed   35
1    Ali   17
2   Omar   25
```

```
In [11]: DataFrame1[1:]
```

Out[11]:

| | Name | Age |
|---|---|---|
| 1 | Ali | 17 |
| 2 | Omar | 25 |