CANDIDATE'S DECLARATION

I hereby declare that I have undertaken the industrial training at **"WEBTEK LABS"** in partial fulfillment of requirements for the degree of **B.TECH (COMPUTER SCIENCE & ENGINEERING)** at **NETAJI SUBHASH ENGINEERING COLLEGE, KOLKATA**. The work which is being presented in the training report submitted to department of **COMPUTER SCIENCE & ENGINEERING** at **NETAJI SUBHASH ENGINEERING COLLEGE, KOLKATA** is an authentic record of training work.

Student Name — Signature of the Student

Amit Raj Singh

The four weeks industrial training Viva-Voice examination of _____ has been held on _____ and accepted.

Signature of Internal Examiner — Signature of External Examiner

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them. I owe my deep gratitude to our project guide **Ms. Mousita Dhar**, who took keen interest on my project work and guided me all along, till the completion of my project work by providing all the necessary information for developing a good project. I heartily thank our internal project guide, **Mr. Malay Mitra**, for his guidance and suggestions during this project work. I am thankful to and fortunate enough to get constant encouragement, support and guidance from all teaching staffs of **Webtek Labs** which helped me in successfully completing my project work. Also, I would like to extend my sincere esteems to all staff in laboratory for their timely support.

Amit Raj Singh

Sem: 6^{th} Year: 3^{rd}

Computer Science & Engineering

CERIFICATE OF APPROVAL

This is to certify that **AMIT RAJ SINGH** (Univ. Roll:- 10900117115) student of **Computer Science & Engineering Department**, **Netaji Subhash Engineering College, Kolkata** has completed his Industrial Training at "**WEBTEK LABS PVT. LTD.** " and submitted this project on "**Survival Prediction in Titanic Disaster**" in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. This is a record of student's own study carried under my supervision & guidance. It is understood that by this approval the undersigned accepts this project only for the purpose for which it is submitted.

Ms. Mousita Dhar

(Project In Charge)

CONTENTS

INTRODUCTION

1.1 Python: Python is a clear and powerful object-oriented programming language, comparable to Pearl, Ruby, Scheme or Java. . It was created by Guido van Rossum, and released in 1991.

- **Python Features:**
    - **Easy to Learn and Use:** Python is easy to learn and use. It is a developer-friendly and high level programming language.
    - **Expressive Language:** Python language is more expressive means that it is more understandable and readable. It Uses an elegant syntax, making the programs we write easier to read.
    - **Interpreted Language:** Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.
    - **Cross-platform Language**: Python can run easily on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language. i.e it is a platform independent language.
    - **Free and Open Source:** Python language is freely available. The source-code is also available. Therefore it is open source and can be modified and its source code can be appended with new features and modules. Python can be freely modified and re-distributed, because while the language is copyrighted it's available under an open source license.
    - **Object-Oriented Language:** Python supports object oriented programming and concepts of classes and objects come into existence with classes and multiple inheritances.

- **Large Standard Library:** Python has a large and broad library and provides rich set of module and functions for rapid application development. Comes with a large standard library that support many common programming task such as connecting to web servers, searching text with regular expressions, reading and modifying files.
- **GUI Programming Support:** Graphical user interfaces can be developed using Python. Libraries like Tkinter are available to design GUIs.
- **Some Other Features:**-
- Python's interactive mode makes it easy to test short snippets of code. There's also a bundled development environment called **IDLE.**
- Is easily extended by adding new modules implemented in a complied language such as C or C++.
- Can also be embedded into an application to provide a programmable interface.
- Code can be grouped into modules and packages.
- The language supports raising and catching exception, resulting in cleaner error handling.
- Data types are strongly and dynamically typed. Mixed incompatible type (e.g. attempting to add a string and a number) causes an exception to be raised, so errors are caught sooner.
- Python contains advanced programming features such as generators and list comprehension.

- **Python Versions:**
  - First released in 1991.

- Python 2.0 was released on 16th October 2000.
- Python 3.0 was released on 3 December 2008.
- 2.7.14 was released on 2017.
- 3.8(current version)

- **Application Of Python:**
  - Web Development
  - Data Analysis
  - Machine Learning
  - Internet of Things
  - GUI Development
  - Image processing
  - Data visualization
  - Game Development

- **1.2 Anaconda:** Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and MacOS.The open-source Anaconda Distribution is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X with over 15 million users worldwide.

- **Anaconda's Features:** It is the industry standard for developing, testing, and training on a single machine, enabling individual data scientists to:
    - Quickly download 1,500+ Python/R data science packages.
    - Manage libraries, dependencies, and environment with Conda.
    - Develop and train machine learning and deep learning models with scikit-learn Tensor Flow, and Theano.
    - Analyze data with scalability and performance with Dask, Numpy, pandas and Numba.
    - Visualize results with Matplot.lib, Bokeh, Datashader, and holoviews.

APPLICATIONS OF ANACONDA

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

- JupyterLab
- Jupyter Notebook
- Qt Console
- Spyder
- Glueviz
- Orange3

- **IPython:** Python (Interactive Python) is a command shell for interactive computing in multiple programming language, originally developed for the Python programming language, that offers introspection, rich media, shell syntax, tab completion, and history.

- **IPython Features:**
    - Interactive shells (terminals and Qt-based).
    - A browser-based notebook interface with support for code, text, mathematical.
    - Expressions, inline plots and other media.
    - Support for interactive data visualization and use of GUI toolkits.
    - Flexible, embeddable interpreters to load into one's own projects.
    - Tools for parallel computing.

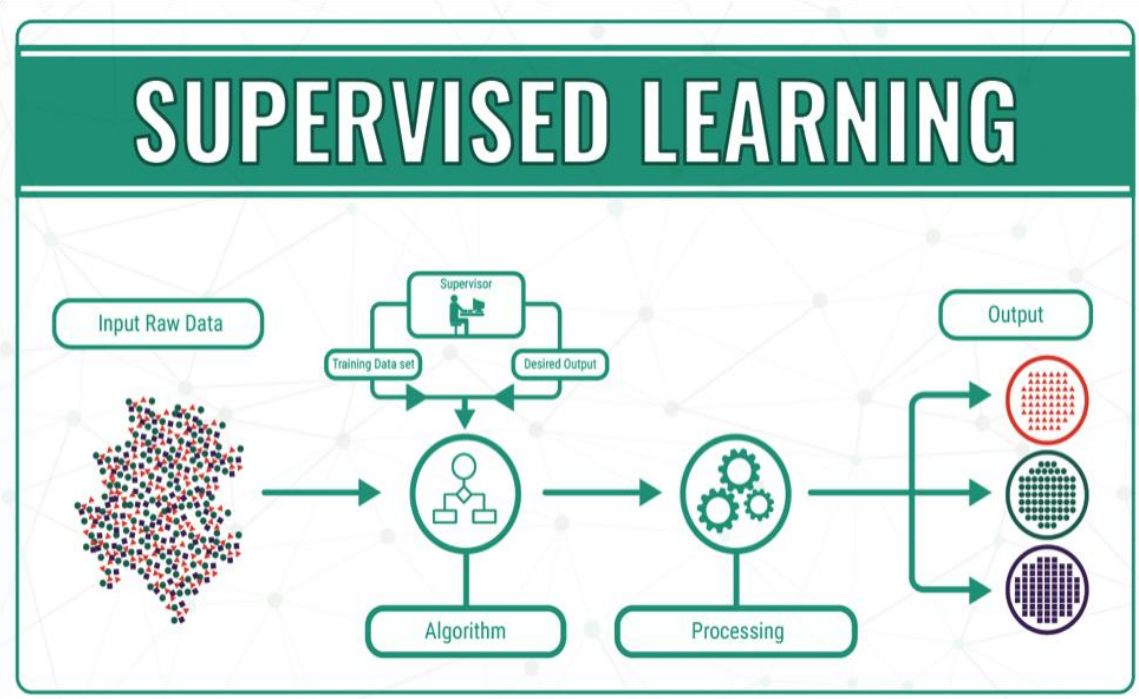
INTRODUCTION TO MACHINE LEARNING

Machine learning is an application of artificial intelligence (AI) that provides system 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. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for pattern in data and make better decision in the future based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly.

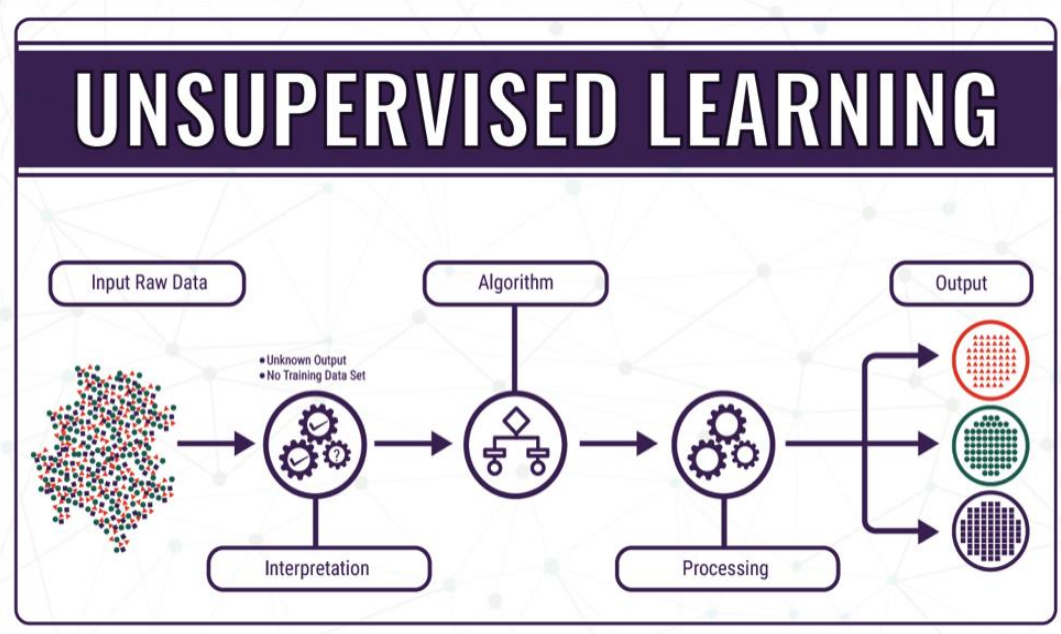
Types of Machine learning:-

Although a machine learning model may apply a mix of different techniques, the method for learning can typically be categorized as three general types:-

- **Supervised learning:** The learning algorithm is given labeled data and the desired output. For example, pictures of dogs labeled “dog” will help the algorithm identify the rules to classify pictures of dogs. Supervised learning is the most common sub branch of machine learning today. Typically, new machine learning practitioners will begin their journey with supervised learning algorithms. Therefore, the first of this three post series will be about supervised learning. Supervised machine learning algorithms are designed to learn by example. The name “supervised” learning originates from the idea that training this type of algorithm is like having a teacher supervise the whole process.



- **<u>Unsupervised learning:</u>** The data given to the learning algorithm is unlabeled, and the algorithm is asked to identify patterns in the input data. For example, the recommendation system of an e-commerce website where the learning algorithm discovers similar items often bought together.



- **Reinforcement learning:** The algorithm interacts with a dynamic environment that provides feedback in term of rewards and punishment. For example, self -driving cars being rewarded to stay on the road.

Applications:

- Handwriting Recognition
- Medical Diagnosis
- Email Spam Filtering
- Recommendation Engine
- Face Detection
- Fraud Detection

Introduction To Numpy And Pandas

- **Numpy:** NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

- **Features of Numpy:-**

    - Numpy (Numeric Python) is a linear algebra library for python.
    - Numpy enriches the programming language Python with powerful data structure for efficient computation of multi-dimensional arrays and matrices.
    - A Numpy array is a grid of values, all of the same types, and is indexed by a tuple of nonnegative integers.
    - The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension.

- **Pandas:** Pandas is an open source library that allows to you perform data manipulation in Python. Pandas library is built on top of Numpy, meaning Pandas needs Numpy to operate. Pandas provide an easy way to create, manipulate and wrangle the data. Pandas is also an elegant solution for time series data.
- **Features of Pandas:-**
    - Pandas is the most popular python library that is used for data analysis.
    - It provides highly optimized performance with back-end source code is purely written in C or Python.
    - We analysis data in pandas with:
        - Series (1-d array)
        - Data Frame (2-d array)
    - Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data – load, prepare, manipulate, model and analyze.

Scikit learn and Tkintertable

SCIKIT-LEARN: Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy

Scikit-learn is largely written in Python, and uses numpy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM: Logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as matplotlib and plotly for plotting, numpy for array vectorization, pandas dataframes, scipy, and many more.

TKINTERTABLE: This set of classes allows interactive spreadsheet-style tables to be added into an application.

This program is free software and can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

Sub-Modules:-

~tkintertable.App
Sample App to illustrate table functionality.

~tkintertable.Custom
Custom Table sub-class illustrate table functionality.

~tkintertable.Dialogs
Table Dialog classes.

~tkintertable.Filtering
Module implements Table filtering and searching functionality.

~tkintertable.Plot
Module for basic plotting inside the TableCanvas. Uses matplotlib.

~tkintertable.Prefs
Manages preferences for Table class.

~tkintertable.TableFormula
Module implements the Formula class for cell formulae.

~tkintertable.TableModels
Module implementing the TableModel class that manages data for it's associated TableCanvas.

~tkintertable.Table_images
Images stored as PhotoImage objects, for buttons and logos.

~tkintertable.Tables
Implements the core TableCanvas class.

~tkintertable.Tables_IO
Import and export classes.

~tkintertable.Testing
Table Testing module.

Steps Of Machine Learning

To apply the learning process to real world tasks, we'll be a live step process. Regardless of the task at hand, any machine learning algorithm can be deployed by following these steps:

- **Data Collection:** The data collection step Involves gathering the learning material an algorithm will use to generate actionable knowledge. In most cases, the data will need to be combined into a single source like a text file, spreadsheet, or database.
- **Data exploration and preparation:** The quality of any machine learning project is based largely on the quality of its input data. Thus, it is important to learn more about the data and its nuances during a practice called data exploration. Additional work is required to prepare the data for the learning process. This involves fixing or cleaning so-called "messy" data, eliminating unnecessary data, and recoding the data to confirm to the learner's expected inputs.
- **Model training:** By the time the data has been prepared for analysis, you are likely to have a sense of what you are capable of learning from the data. The specific machine learning task chosen will inform the selection of an appropriate algorithm, and the algorithm will represent the data in the form of a model.
- **Model evaluation:** Because each machine learning model results in a biased solution to the learning problem, it is important to evaluate how well the algorithm learns from its experience. Depending on the type of model used, you might be able to evaluate the accuracy of the model using a test dataset or you may need to develop measures of performance specific to the intended application.

- **Model improvement:** If better performance is needed, it becomes necessary to utilize more advanced strategies to augment the performance of the model. Sometimes, it may be necessary to switch to a different type of model altogether. You may need to supplement your data with additional data or perform additional preparatory work as in step two of this process.

Supervised Learning

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that supervised learning algorithm analyses the training data set of training examples) and produces a correct outcome from labeled data.

For instance, suppose we are given a basket filled with different kinds of fruits. Now the first step is to train the machine with all different fruits one by one like this:

- If shape of object is rounded and depression at top having color Red then it will be labeled as - Apple.
- If shape of object is long curving cylinder having color Green-Yellow then it will be labeled as - Banana.

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

Since the machine has already learned the things from previous data and this time have to use it wisely. It will first classify the fruit with its shape and color and would confirm the fruit name as BANANA and put it in Banana category. Thus the machine learns the things from training data (basket containing fruits) and then apply the knowledge to test data (new fruit). Supervised Learning is where we have input

variables(x) and an output variable(y) and we use an algorithm to learn the mapping function from the input to the output.

Process Flow: Supervised Learning:

Supervised learning classified into two categories of algorithms:

- **Classification:** A classification problem is when the output variable is a category, such as "Red" or "blue" or "disease" and "no disease".
    - A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes.
    - For example, when filtering emails "spam" or "not spam", when looking at transaction data, "fraudulent", or "authorized".

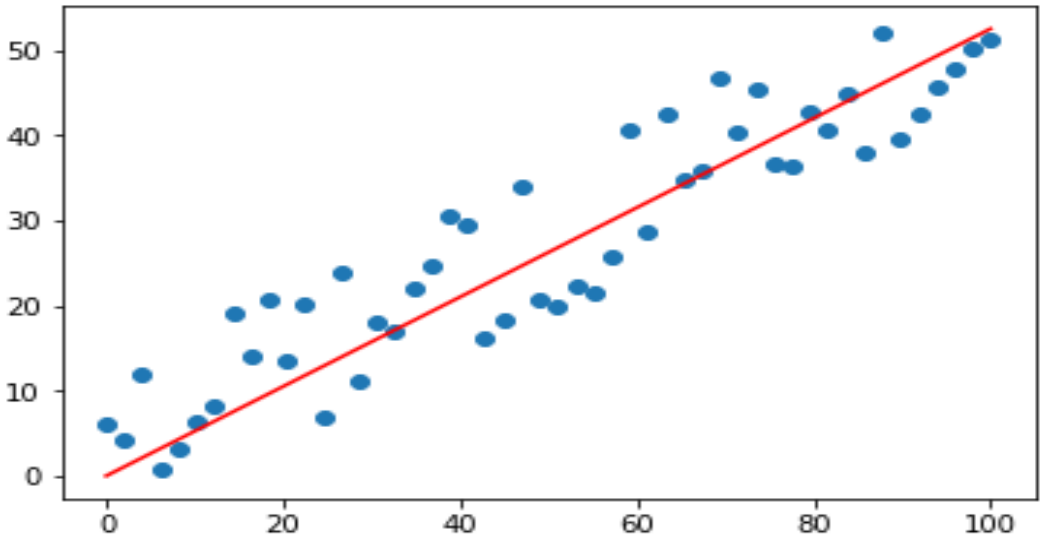
Regression: A regression problem is when the output variable is a real or continuous value, such as "dollars" or "weight" or "salary".

- **Linear Regression:** It is also called simple linear regression. It establishes the relationship between two variables using a straight line. Linear regression attempts to draw a line that comes closest to the data by finding the slope and intercept that define the line and minimize regression errors.

  If two or more explanatory variables have a linear relationship with the dependent variable, the regression is called a multilinear regression.

  Many data relationships do not follow a straight line, so statisticians use nonlinear regression instead. The two are similar

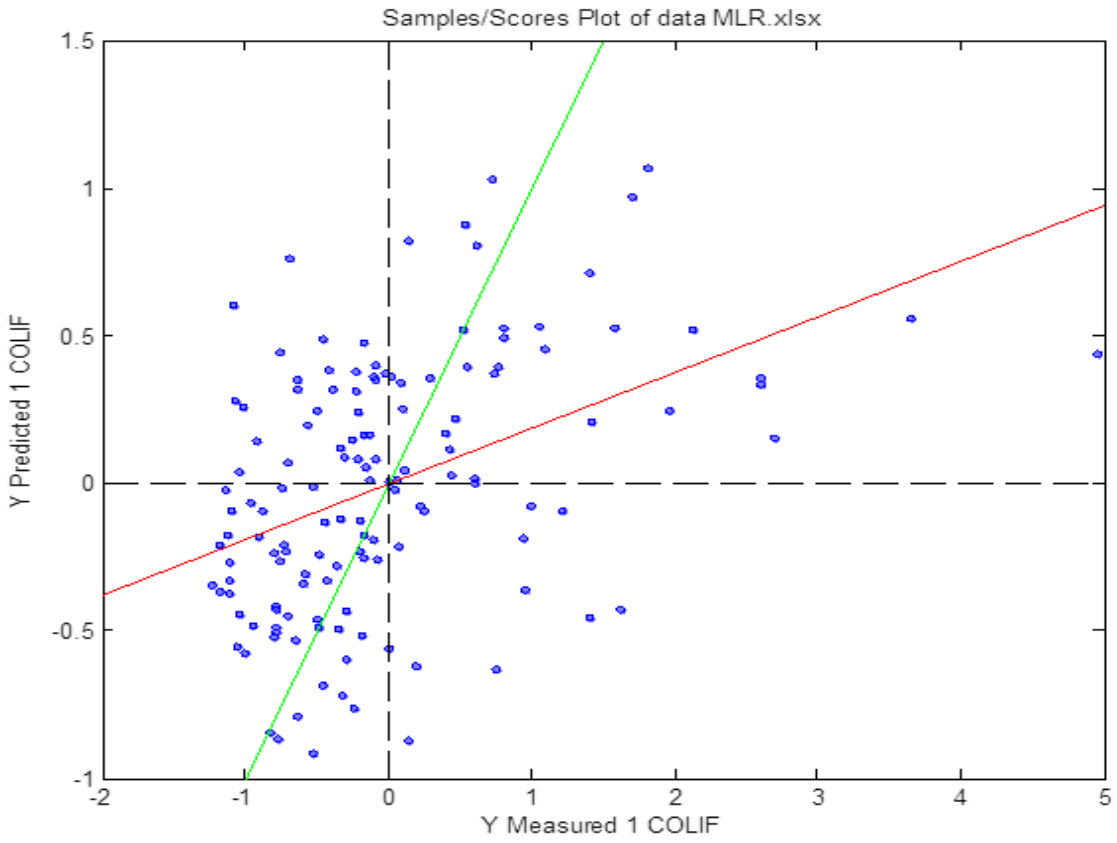
in that both track a particular response from a set of variables graphically. But nonlinear models are more complicated than linear models because the function is created through a series of assumptions that may stem from trial and error. It tries to fit data with the best hyper-plane which goes through the points.



- **Multiple Regression:** It is rare that a dependent variable is explained by only one variable. In this case, an analyst uses multiple regression, which attempts to explain a dependent variable using more than one independent variable. Multiple regressions can be linear and nonlinear.

Multiple regressions are based on the assumption that there is a linear relationship between both the dependent and independent variables. It also assumes no major correlation between the independent variables.

As mentioned above, there are several different advantages to using regression analysis. These models can be used by businesses and economists to help make practical decisions.



Training Work Undertaken

Titanic Survival Prediction Problem

In **"Titanic Survival Prediction Problem"** problem we are asked to predict whether a passenger on the titanic ship would have survived or not. Based on many factors like passenger class, sex, age, point of embarks etc.

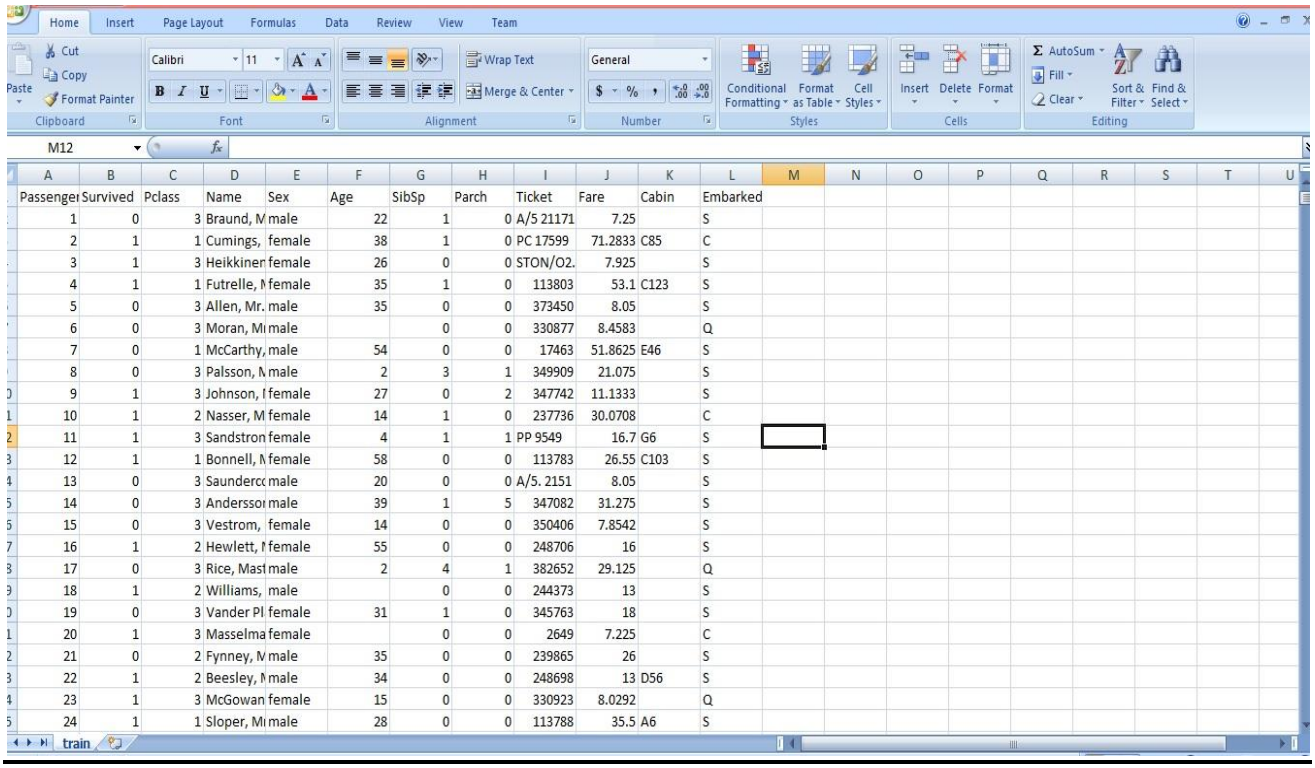
Aim: Our aim was to predict whether a passenger on the titanic would have been survived or not.

Dataset Features:-

The dataset contains 1310 records and 9 Features. The features of the dataset are given below:

1. PClass – Passenger Class
2. Survived
3. Name
4. Sex
5. Age
6. Sibsp
7. Parch
8. Ticket
9. Embarked

Dataset Snapshot:-



PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings,	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen	female	26	0	0	STON/O2.	7.925		S
4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mi	male		0	0	330877	8.4583		Q
7	0	1	McCarthy,	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, M	male	2	3	1	349909	21.075		S
9	1	3	Johnson, I	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, M	female	14	1	0	237736	30.0708		C
11	1	3	Sandstron	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, M	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercc	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Anderssor	male	39	1	5	347082	31.275		S
15	0	3	Vestrom,	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, M	female	55	0	0	248706	16		S
17	0	3	Rice, Mast	male	2	4	1	382652	29.125		Q
18	1	2	Williams,	male		0	0	244373	13		S
19	0	3	Vander Pl	female	31	1	0	345763	18		S
20	1	3	Masselma	female		0	0	2649	7.225		C
21	0	2	Fynney, M	male	35	0	0	239865	26		S
22	1	2	Beesley, M	male	34	0	0	248698	13	D56	S
23	1	3	McGowan	female	15	0	0	330923	8.0292		Q
24	1	1	Sloper, Mi	male	28	0	0	113788	35.5	A6	S

- **Data Visualization:**
  - Data visualization is the discipline of trying to understand data by placing it in a visual context.
  - Python offers multiple great graphing libraries that come packed with lots of different features.
  - Can be done with the help of:
    - Matplot.lib
    - Seaborne

- **Matplot.lib:**
- Matplot.lib is the most popular python plotting library.
- It is a low-level library with a Mat lab like interface which offers lots of freedom at the cost of having to write more code.
- Matplot.lib is specifically good for creating basic graphs like line charts, bar charts, histograms and many more.
- Can be imported as Import matplot.lib pie plot as plat.
- Different plots in matplot.lib:
    - Scatter plot
    - Plot
    - Histogram
    - Bar chart

- **<u>Evaluation:</u>**
    - **titanic file on pandas df (conversion from xls to csv)**

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#ignore warnings
import warnings
warnings.filterwarnings('ignore')

titan_df = pd.read_excel(r'titanic.xls')
titan_df.head()
```

Out[4]:

	pclass	survived	name	sex	age	sibsp	parch	ticket	embarked
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	S
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	S
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	S
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	S
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	S

Report on panda DF

1. **Data Wrangling:** After creating the dataframe, we performed certain operations on the datasets to eliminate the missing data using isnull(), info() and dropna(). These are the part of the data pre-processing which means making the data ready for applying certain Machine learning algorithms.

In [5]:
```
print('Titan DF status to find missing values:')
print(titan_df.isnull().sum())
print('Titan DF info output')
print(titan_df.info())
print('Deleting Missing Data')
titan_df = titan_df.dropna(axis = 0, how ='any')
print('Titan DF info output after deleting missing data')
print(titan_df.isnull().sum())
print(titan_df.info())
```

```
Titan DF status to find missing values:
pclass        0
survived      0
name          0
sex           0
age         263
sibsp         0
parch         0
ticket        0
embarked      2
dtype: int64
Titan DF info output
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 9 columns):
pclass      1309 non-null int64
survived    1309 non-null int64
name        1309 non-null object
sex         1309 non-null object
age         1046 non-null float64
sibsp       1309 non-null int64
parch       1309 non-null int64
ticket      1309 non-null object
embarked    1307 non-null object
dtypes: float64(1), int64(4), object(4)
memory usage: 92.1+ KB
None
```

2. **Survived vs Non-survived:**

In [7]:
```
s = titan_df["survived"].value_counts()
print('Not Survived of Total passengers:', s[0])
print('Survived of Total passengers:', s[1])

s= titan_df['survived'].value_counts(normalize = True)
print(s[0], s[1])
print('Percentage of Total Non Survival : %.2f' % (s[0] * 100))
print('Percentage of Total Survival : %.2f'  % (s[1] * 100))
```

```
Not Survived of Total passengers: 619
Survived of Total passengers: 425
0.592911877394636 0.407088122605364
Percentage of Total Non Survival : 59.29
Percentage of Total Survival : 40.71
```

3. Females Survived vs Non-Survived:

In [11]:
```
s = titan_df['survived'][titan_df['sex'] == 'female'].value_counts()
print('Total females not survived :', s[0])
print('Total females survived :', s[1])
s = titan_df['survived'][titan_df['sex'] == 'female'].value_counts(normalize = True)
print(s)
print()
print('Percentage of females not survived : %.2f' % (s[0]*100))
print('Percentage of females survived : %.2f' % (s[1]*100))
```

```
Total females not survived : 96
Total females survived : 290
1    0.751295
0    0.248705
Name: survived, dtype: float64

Percentage of females not survived : 24.87
Percentage of females survived : 75.13
```

4. Normalized survival rates for passengers under 18 and above 18:

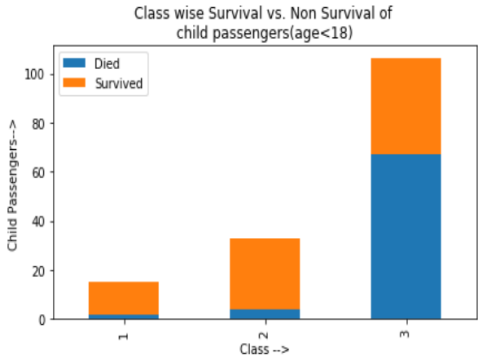
In [15]:
```
titan_df["child"] = float('NaN')
titan_df['child'][titan_df['age'] < 18] = 1
titan_df['child'][titan_df['age'] >= 18] = 0

# Print normalized Survival Rates for passengers under 18
s = titan_df["survived"][titan_df["child"] == 1].value_counts(normalize = True)
print(s)
print('Percentage Not Survived under age 18 : %.2f' % (s[0] * 100))
print('Percentage Survived under age 18 : %.2f' % (s[1] * 100))

# Print normalized Survival Rates for passengers 18 or older (>= 18)
s = titan_df["survived"][titan_df["child"] == 0].value_counts(normalize = True)
print(s)
print('Percentage Not Survived for age >= 18 : %.2f' % (s[0] * 100))
print('Percentage Survived for age >= 18 : %.2f' % (s[1] * 100))
```

```
1    0.525974
0    0.474026
Name: survived, dtype: float64
Percentage Not Survived under age 18 : 47.40
Percentage Survived under age 18 : 52.60
0    0.613483
1    0.386517
Name: survived, dtype: float64
Percentage Not Survived for age >= 18 : 61.35
Percentage Survived for age >= 18 : 38.65
```

5. Classwise Survival vs Non-Survival of Child Passengers:

In [16]:
```
print('Class wise Survival vs. Non Survival of \nchild passengers(age<18)')
print(titan_df[titan_df.child==1].groupby(['pclass', 'survived']).size())
titan_df[titan_df.child==1].groupby(['pclass','survived']).size().unstack().plot(kind='bar', stacked=True)
plt.title('Class wise Survival vs. Non Survival of \nchild passengers(age<18)')
plt.tight_layout()
plt.legend(['Died','Survived'])
plt.xlabel('Class -->')
plt.ylabel('Child Passengers-->')
plt.show()
```

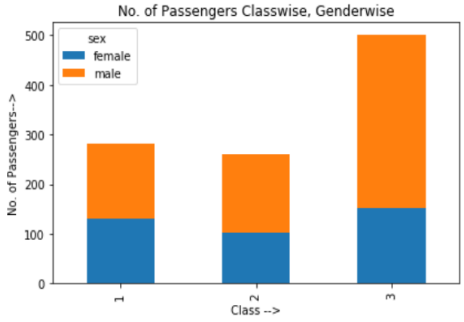
```
Class wise Survival vs. Non Survival of
child passengers(age<18)
pclass  survived
1       0           2
        1          13
2       0           4
        1          29
3       0          67
        1          39
dtype: int64
```



6. Number of Passengers classwise, genderwise:

In [17]:
```
# No of passengers class wise, genderwise
print('No. of passengers class wise, gender wise')
print(titan_df.groupby(['pclass', 'sex']).size())
titan_df.groupby(['pclass', 'sex']).size().unstack().plot(kind='bar', stacked=True)
plt.title('No. of Passengers Classwise, Genderwise')
plt.xlabel('Class -->')
plt.ylabel('No. of Passengers-->')
plt.tight_layout()
plt.show()
```

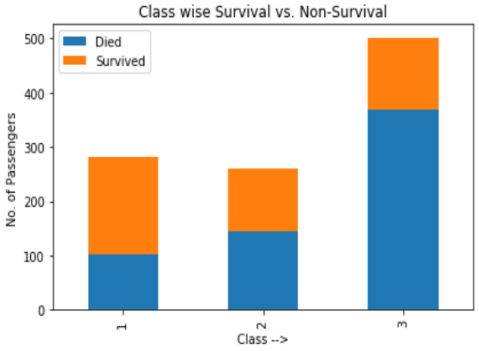
```
No. of passengers class wise, gender wise
pclass  sex
1       female    131
        male      151
2       female    103
        male      158
3       female    152
        male      349
dtype: int64
```



7. Classwise Survival vs Non-Survival:

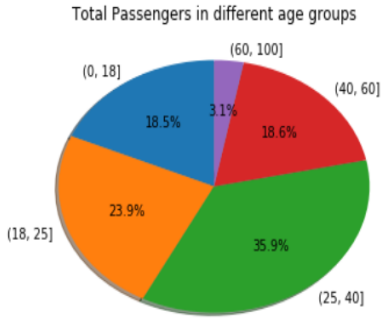
In [18]:
```
# Class wise survival vs. non-survival
print('Class wise Survival vs. Non-Survival')
print(titan_df.groupby(['pclass', 'survived']).size())
titan_df.groupby(['pclass', 'survived']).size().unstack().plot(kind='bar', stacked=True)
plt.title('Class wise Survival vs. Non-Survival')
plt.xlabel('Class -->')
plt.ylabel('No. of Passengers')
plt.legend(['Died', 'Survived'])
plt.tight_layout()
plt.show()
```

```
Class wise Survival vs. Non-Survival
pclass  survived
1       0           103
        1           179
2       0           146
        1           115
3       0           370
        1           131
dtype: int64
```



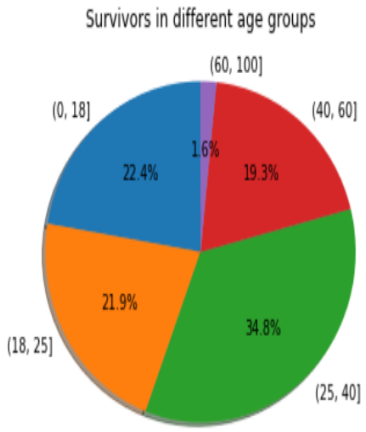
8. Pie-Chart of passengers age groups:

In [20]:
```
# Pie chart
age_bin = [0, 18, 25, 40, 60, 100]     # defined age bin intervals
# create the bins
titan_df['AgeBin'] = pd.cut(titan_df.age, bins=age_bin)
d_temp = titan_df[np.isfinite(titan_df['age'])]    # removing null rows
# Number of survivors based on age bin
survivors = d_temp.groupby('AgeBin')['survived'].agg(sum)
# Total passengers in each bin
total_passengers = d_temp.groupby('AgeBin')['survived'].agg('count')
# Plot pie chart
plt.pie(total_passengers, labels=total_passengers.index.values.tolist(), autopct='%1.1f%%', shadow=True, startangle=90)
plt.title('Total Passengers in different age groups')
plt.show()
```



9. Pie-Chart of Passengers in different age groups:-

In [21]:
```
plt.pie(survivors, labels=survivors.index.values.tolist(), autopct='%1.1f%%', shadow=True, startangle=90)
plt.title('Survivors in different age groups')
plt.show()
```



Discussions

“Titanic survival prediction” problem we are asked to find whether a passenger on the titanic survived or not. During this project we were given titanic data set and by various methods we had to cleanup the data and plot the different ratios and graphs as asked in the project workup. We did the representation using matplotlib, pandas, numpy and pyplot.

Conclusion

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future, based on the examples that we provide. The primary aim is to allow the computers learn automatically without human intervention or assistance and adjust actions accordingly. Our aim by “titanic survival prediction" is to try to and analyze the number of survivals of different gender and different classes and across different age groups.

References

1. https://community.alteryx.com/t5/Data-Science-Blog/Life-or-Death-Prediction-with-the-Titanic-Dataset/ba-p/178966
2. https://towardsdatascience.com/a-beginners-guide-to-python-for-data-science-60ef022b7b67
3. https://en.wikipedia.org/wiki/Machine_learning