**Homework Assignment 1  
Machine Learning (CS-596)**

**Name: Dhaval Harish Sharma  
Red ID: 824654344**

**Question 1**

In python, the library [matplotlib.pyplot](https://matplotlib.org/api/pyplot_api.html#module-matplotlib.pyplot) is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc. In [matplotlib.pyplot](https://matplotlib.org/api/pyplot_api.html#module-matplotlib.pyplot) various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes (please note that “axes” here and in most places in the documentation refers to the *axes* [part of a figure](http://matplotlib.org/faq/usage_faq.html#parts-of-a-figure) and not the strict mathematical term for more than one axis).

Please select a proper range for the variable x, and visualize the following functions:

* Straight line where
* Quadratic function:
* Log function, and ,
* Sigmoid function,

**(1) Straight Line:**

**Source Code:**   
#Importing the required libraries  
import numpy as np  
import matplotlib.pyplot as plt

#Creating a new figure  
plt.figure()

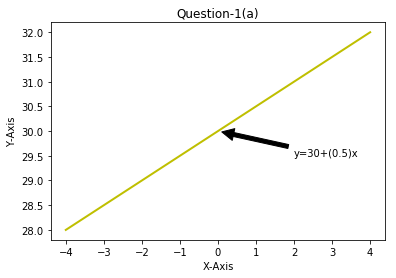
#Initializing x and y values  
x\_values = np.linspace(-4, 4, 5)  
y\_values = (0.5 \* x\_values) + 30

#Plotting the values of x and y in the output  
plt.plot(x\_values, y\_values, '-y', lw = 2)

#Annotating the graph  
plt.annotate('y=30+(0.5)x', xy=(0, 30), xytext=(2, 29.5), arrowprops = dict(facecolor='black', shrink=0.05))

#Inserting proper labels and title  
plt.xlabel('X-Axis')  
plt.ylabel('Y-Axis')  
plt.title('Question-1(a)')

#Displaying the graph in the output  
plt.show()

**Output:  
  
(2)** **Quadratic Function:  
Source Code:**#Importing the required libraries  
import numpy as np  
import matplotlib.pyplot as plt

#Creating a new figure  
plt.figure()

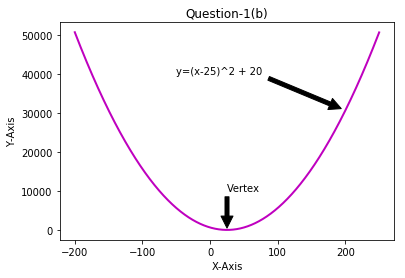
#Initializing x and y values  
x\_values = np.arange(-200, 250, 0.1)  
y\_values = (x\_values - 25)\*\*2 + 20

#Plotting the values of x and y in the output  
plt.plot(x\_values, y\_values, '-m', lw = 2)

#Annotating the graph  
plt.annotate('y=(x-25)^2 + 20', xy=(200, 30645), xytext=(-50, 40000), arrowprops = dict(facecolor='black', shrink=0.05))  
plt.annotate('Vertex', xy=(25, 20), xytext=(25, 10000), arrowprops = dict(facecolor='black', shrink=0.05))

#Inserting proper labels and title  
plt.xlabel('X-Axis')  
plt.ylabel('Y-Axis')  
plt.title('Question-1(b)')

#Displaying the graph in the output  
plt.show()

**Output:**  


**(3)** **Log Function:  
Source Code:**  
#Importing the required libraries  
import numpy as np  
import matplotlib.pyplot as plt

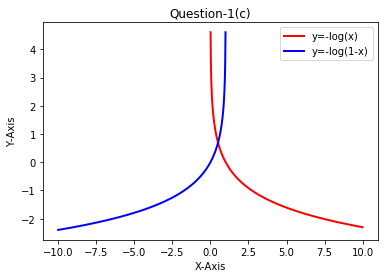
#Creating a new figure  
plt.figure()

#Initializing x and y values  
x\_values1 = np.arange(0.01, 10, 0.01)  
y\_values1 = -(np.log(x\_values1))  
x\_values2 = np.arange(-10, 1, 0.01)  
y\_values2 = -(np.log(1 - x\_values2))

#Plotting the values of x and y in the output  
plt.plot(x\_values1, y\_values1, '-r', label = 'y=-log(x)', lw = 2)  
plt.plot(x\_values2, y\_values2, '-b', label = 'y=-log(1-x)', lw = 2)

#Inserting proper labels, title and legend  
plt.xlabel('X-Axis')  
plt.ylabel('Y-Axis')  
plt.title('Question-1(c)')  
plt.legend()

#Displaying the graph in the output  
plt.show()

**Output:**  


**(4)** **Sigmoid Function:  
Source Code:**#Importing the required libraries  
import numpy as np  
import matplotlib.pyplot as plt

#Creating a new figure  
plt.figure()

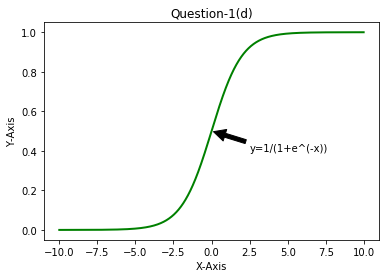
#Initializing x and y values  
x\_values = np.arange(-10, 10, 0.01)  
y\_values = 1 / (1 + np.exp(-x\_values))

#Plotting the values of x and y in the output  
plt.plot(x\_values, y\_values, '-g', label = 'y=-log(x)', lw = 2)

#Annotating the graph  
plt.annotate('y=1/(1+e^(-x))', xy=(0, 0.5), xytext=(2.5, 0.4), arrowprops = dict(facecolor='black', shrink=0.05))

#Inserting proper labels and title  
plt.xlabel('X-Axis')  
plt.ylabel('Y-Axis')  
plt.title('Question-1(d)')

#Displaying the graph in the output  
plt.show()

**Output:  
****Question 2**

2.1 You are given the following tasks, all of which can be solved with a certain type of machine learning algorithms.

1. Classify a house to be single family or townhouse. A training set is available. Each training sample is provided with multiple features, including number of bed rooms, number of bathrooms, and house type (single family or townhouse).
2. Classify an email to be spam or not. Users already identified some emails as spam ones.
3. Human tumor Microarray data are provided as a matrix where rows correspond to genes and columns to tissue samples. The task is to cluster columns (or samples) to identify disease profiles: tissues with similar disease should yield similar expression profiles.

Which statement is correct? (Mark one)

1. i) unsupervised learning with discrete predictions; ii) supervised learning with continuous predictions; iii) supervised learning with continuous predictions;
2. **i) supervised learning with discrete predictions; ii) supervised learning with discrete predictions; iii) unsupervised learning with discrete results;**
3. i) supervised learning with discrete predictions; ii) supervised learning with continuous predictions; iii) unsupervised learning with discrete results;
4. All the three scenarios can be solved by unsupervised learning.

**Question 3**  
Please figure out and describe one machine learning task, either classification or regression, based on your own experiences and background. It should not be the demo tasks taught in class, e.g., hand digit recognition, house pricing, fish classification etc. Please design your machine learning algorithm, and answer the following questions. Note that you only need to submit your plans. No source codes or results are required.

*Task*: Please briefly explain the inputs, outputs, and your goal in general.

*Data preparation*: Please describe how to collect dataset, including training data, validation data, and testing data. Please explain how to get the ground-truth label for all samples;

**Solution:**The machine learning task is to predict the winner of CS:GO matches in the tournaments as well as count the odds for esports betting.

Starting with some of the background, Counter Strike Global Offensive is a first person shooter game whose typical matchmaking consists of five versus five matches. I have been playing this game from the last 4 years. There are two teams called terrorists and counter terrorists along with lots of maps to play. There are several tournaments happening each month consisting of a price pool of millions of dollars. The esports industry is getting bigger day-by-day with games like dota2 whose price pool was almost equal to the fifa world cup. Due to this, the betting systems are upgrading at the same rate which results in a need of a system which can help in betting by coming with a set of outcome consisting of a probability of winner in the upcoming match. The machine learning system can be useful to predict the match winners; also it can be used to count the number of odds for betting online. This system can not only be used for esports but it can also be used for several other sports such as football, cricket, basketball, etc.

The basic features for predicting the winners of a match are past matches, player stats, map win ratio, past matches against the opponent and team rankings. The machine learning algorithm will take inputs from [hltv.org](https://www.hltv.org) as well as some other websites. We can easily get the data of past matches with winners from this website. Also, we can get the data about %win on various maps as well as the player stats from there. In addition to this, there are several websites such as cs.money and hellcase which offers the popularity of various matches as well as past odds on betting. This can also help to predict the winners of the upcoming matches and to calculate the odds which can then be helpful for betting.

The data that is collected can be pre-processed in order to get good results. Then, it can be input to a machine learning model such as multiple linear regression, random forest regression, etc., which will take the inputs from the websites and then calculate the probability of the winner of today’s match. The data that is obtained from [hltv.org](https://www.hltv.org) can be divided in to training set, validation set and the test set with 75%, 10%, 15% of the total data respectively. The system can be trained in multiple epochs until a reasonable amount of accuracy is gained. The training set can contain past matches of the corresponding teams, player stats during a span of two years, team rankings, etc. After gaining a good amount of accuracy, the system can be tested on the validation set. The accuracy gained while testing on the validation set decides whether the system is ready to be tested on the testing data. While testing the model on testing data, the accuracy of predicting the match winner can be compared to the actual match and if the results are reasonable then it can be said that the model is ready.

Thus, after training and testing the model properly, it can be deployed to predict the probability of the upcoming csgo match’s winner. In addition to this, it can also help the people to place a bet on a team with certain amount of accuracy.