HMWK Group 1

Team Members:

1. Rabie Ahmed M AlSaihati
2. Rudrashis Poddar
3. Harsh Kantilal Sharma

University of Cincinnati

College of Applied Science and Engineering

EECS DEPARTMENT – Machine Learning

Professor Anca Ralescu

Homework One

**Problem 1**

The Tic-Tac-Toe learning problem is as follows:

* **Task T :** Playing Tic Tac Toe
* **Performance:** Games won versus humans
* **Experience:** Games played against itself

1) We will begin by assigning an evaluation function that assigns a numerical value to any given board state V: B to R. This function will define the all values which board can take from the set of Real numbers.

2) Target function definition: Vtraining (b) ←100 (if Won) | 0 (if a Draw) | -100 (if Loss)

3) Representation for target function is:

V̂ (b) = w0+w1x1+w2·x2+w3·x3+w4·x4+w5·x5+w6·x6

Following are the Board features:

* x1 is of instances where there are 2 x’s in a row.
* x2 is of instances where there are 2 o’s in a row.
* x3 is of instances where there is an x in a completely open row.
* x4 is of instances where there is an o in a completely open row.
* x5 is of instances of 3 x’s in a row (value of 1 signifies end game)
* x6 is of instances of 3 o’s in a row (value of 1 signifies end game)

4) Assign random hypothesis for weights for example 4 and generate training data by letting it play against another learner with same weights

5) LMS training rule to update weights

Repetitive Process: For random example of boardstate:

Error (b) = Vtraining (b) - V̂ (b)

wi = wi + Lamda\*(Error (b))\*xi, where Lamda = rate of learning & xi = feature vector

Program modules:

1. Performance System: The input is the problem provided by the experiment generator and finally uses the improved learning algorithm to produce a solution trace of the game at every epoch. In our scenario, it is done by the simulation of a Tic-Tac-Toe match.
2. Experiment Generator: This, makes new problem statement at the start of training epoch. Here, it returns an empty initial board state.
3. Critic: The module uses solution trace to output training examples set that are inputted in the generalizer.
4. Generalizers: This is communicates with the critic training examples to update the target function By LMS improvement rule.

**Problem 2**

1-There are 80 legal tic-tac-toe endgame boards

2- There are 9 attributes which are the squares in the Tic-Tac-Toe Board

3- The attribute information is: [X=player X, O = player O, b = blank)



1. top-left-square: {X, O, b}

2. top-middle-square: {X, O, b}

3. top-right-square: {X, O, b}

4. middle-left-square: {X, O, b}

5. middle-middle-square: {X, O, b}

6. middle-right-square: {X, O, b}

7. bottom-left-square: {X, O, b}

8. bottom-middle-square: {X, O, b}

9. bottom-right-square: {X, O, b}

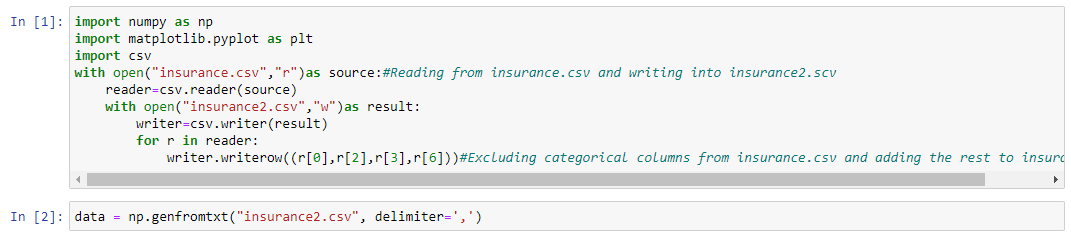
10. Result: {positive, negative}

Result Analysis: 61.25% are positive (i.e., wins for "O")

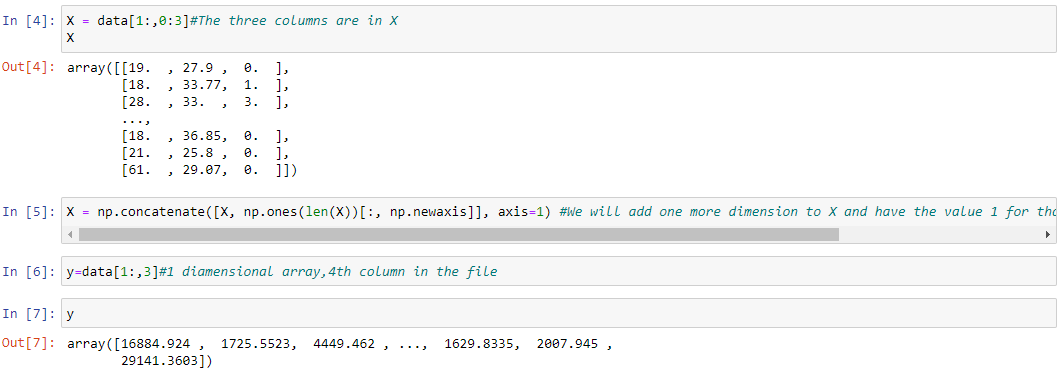
1. As training set increases performance decreases. This is attributed to the search of input in every row of the data.
2. The higher the number of features the lower the performance is decreasing.

**Problem 3**

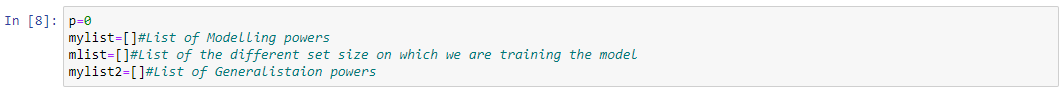
* We start with importing the libraries such as **NumPy** (for mathematical calculations), **Matplotlib** (for visualization) and **CSV** from python for reading and parsing dataset.

****

* Then we capture the **dependent variables in variable X** and **independent variable in variable y**.

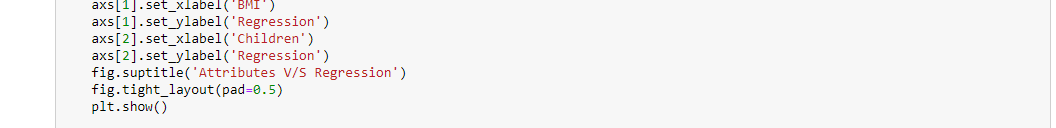
****

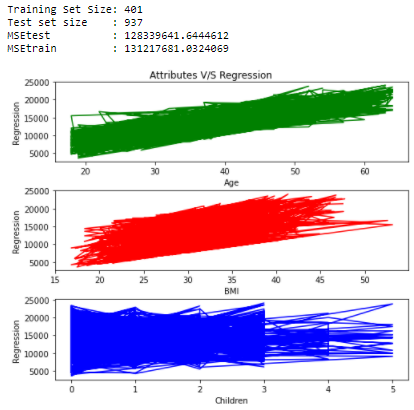
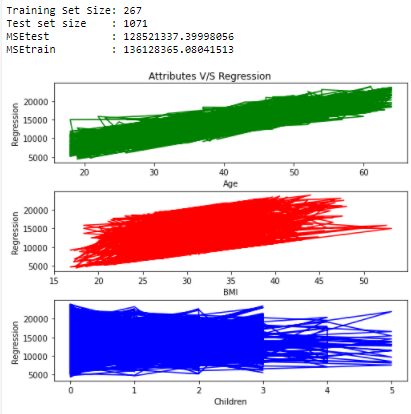
* After this, we have created 3 lists to capture the training set size, generalization powers and modeling powers.

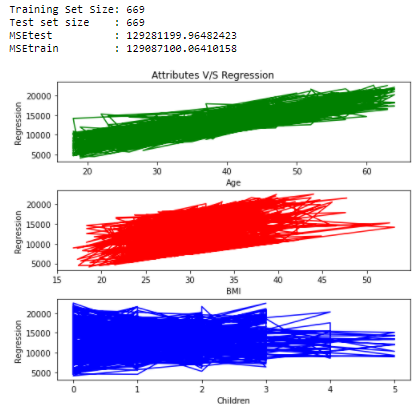
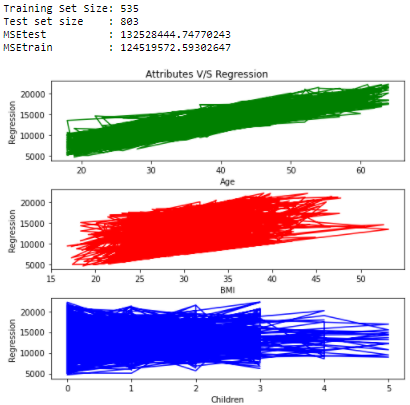
****

* For next step we iterate from 20-80 percent of training set size for the whole code till the attribute plotting as shown below.

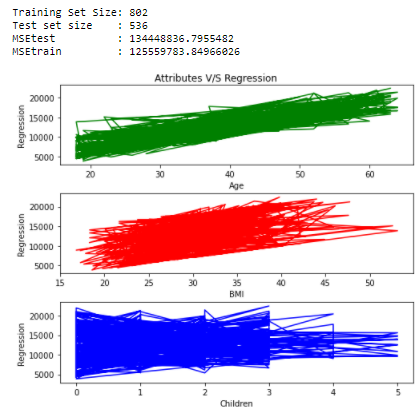
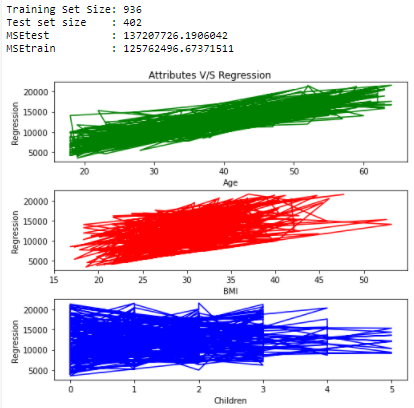
****

****

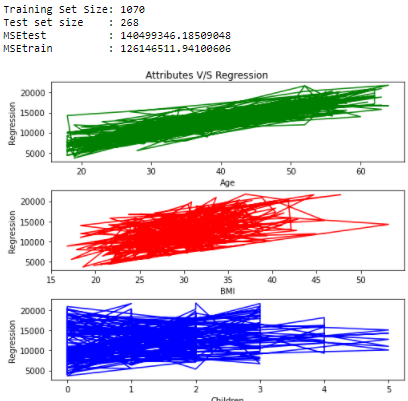
* Below is the plot for training set with 20% & 30% data. ****
* Plot for 40% & 50% training size data.

****

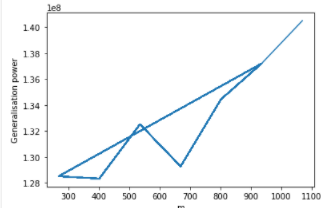
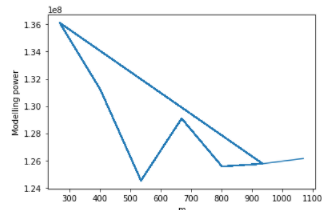
* Plot for 60% & 70% training size data.

****

* Plot for 80% training size data.

****

* Lastly, we’re plotting the Genrelisation and Modelling power plots:

** **