**Dataset** : Adult dataset

**Input Attributes after Pre processing** : 87

**Goal** : To compare the performance of LVQ2, Backpropagation and SVM’s on this dataset

**Steps followed for PreProcessing:**

1. The dataset has the class values (i.e., the output values) in string. Since it is a binary classification (>=50K or <50K) the values in the output class has been changed to +1 and 0 respectively. +1 represents the class with values >=50K and 0 represents the class value <50K.
2. The dataset has few missing values and so the instances of the dataset that has missing values in any parameter is removed.

**traindata=traindata.dropna()**

1. The dataset is split into test data and train data using train test split from sklearn. To make the training dataset, the dataset is split as (x, y) format where x is the input and y being the output.

**from sklearn.model\_selection import train\_test\_split**

**train,test= train\_test\_split(traindata,test\_size=0.3)**

1. The dataset has many categorical values like work class, sex, race, country, education, etc. If we randomly assign values to each category there is a chance that the neural network start forming correlation between those values. To remove this constraint, we are applying one hot encoding on this data so that the neural network doesn’t form any correlation between the values.

**train\_nn\_ip = pd.get\_dummies(train\_ip)**

**test\_nn\_ip = pd.get\_dummies(test\_ip)**

1. To make all the values under a standard scalar, standard scalar library function is used. Once the pre-processing is done this data is fed to the neural network.

**sc = StandardScaler()**

**X\_train = sc.fit\_transform(train\_nn\_ip)**

**X\_test = sc.transform(test\_nn\_ip)**

**Performance of algorithms when learning rate is modified:**

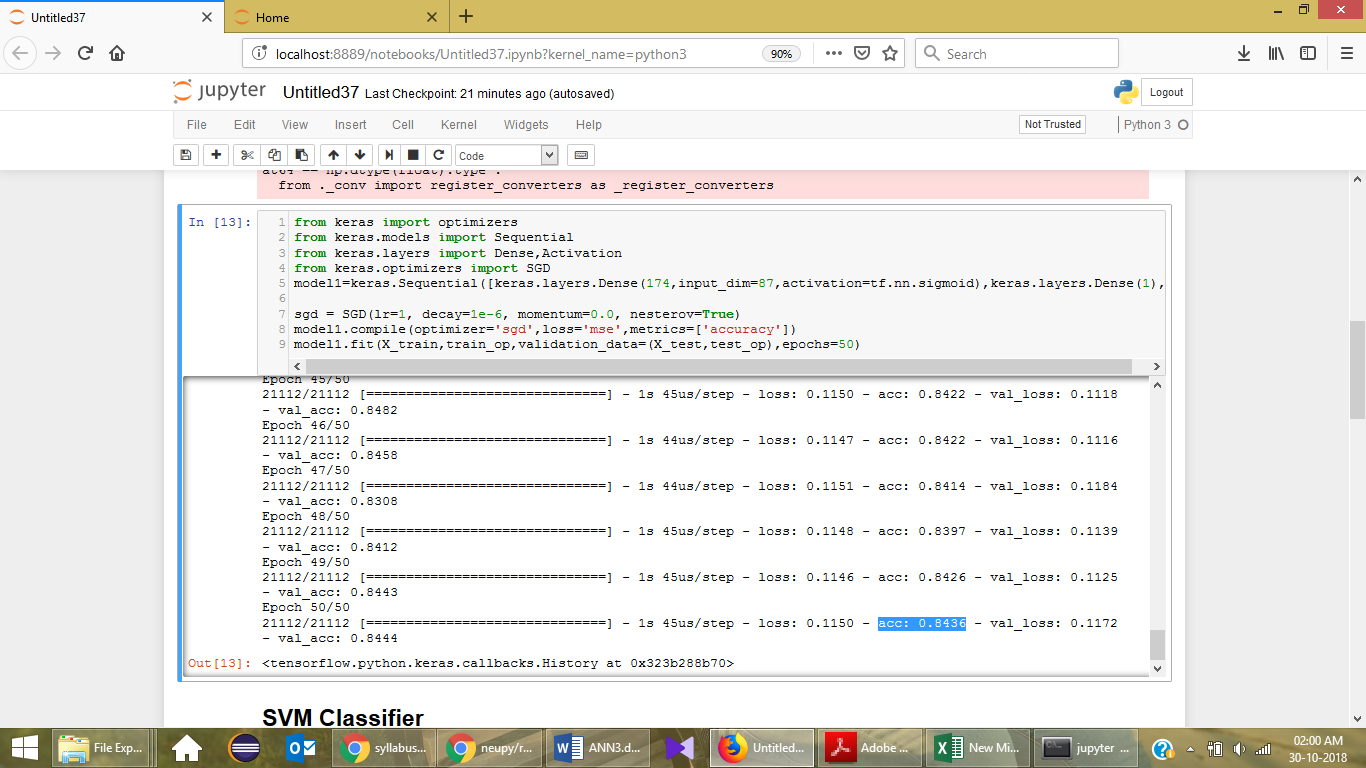
**Backpropagation**:

Eg : When learning parameter in a simpe d-2d-1 neural network architecture, the accuracy value is 0.8436

As in theory, the reduction in led to improvements in accuracy to a certain factor. After that since the convergence to global minimum is altered/made very slow because of the small values the error rate for the back propagation algorithm doesn’t change/reduces after that.

|  |  |
| --- | --- |
| **Learning Rate** | **Back Propagation** |
| 1 | 0.8436 |
| 0.5 | 0.8471 |
| 0.25 | 0.8445 |
| 0.01 | 0.8444 |
| 0.001 | 0.8446 |

Various values for is experimented as shown below:



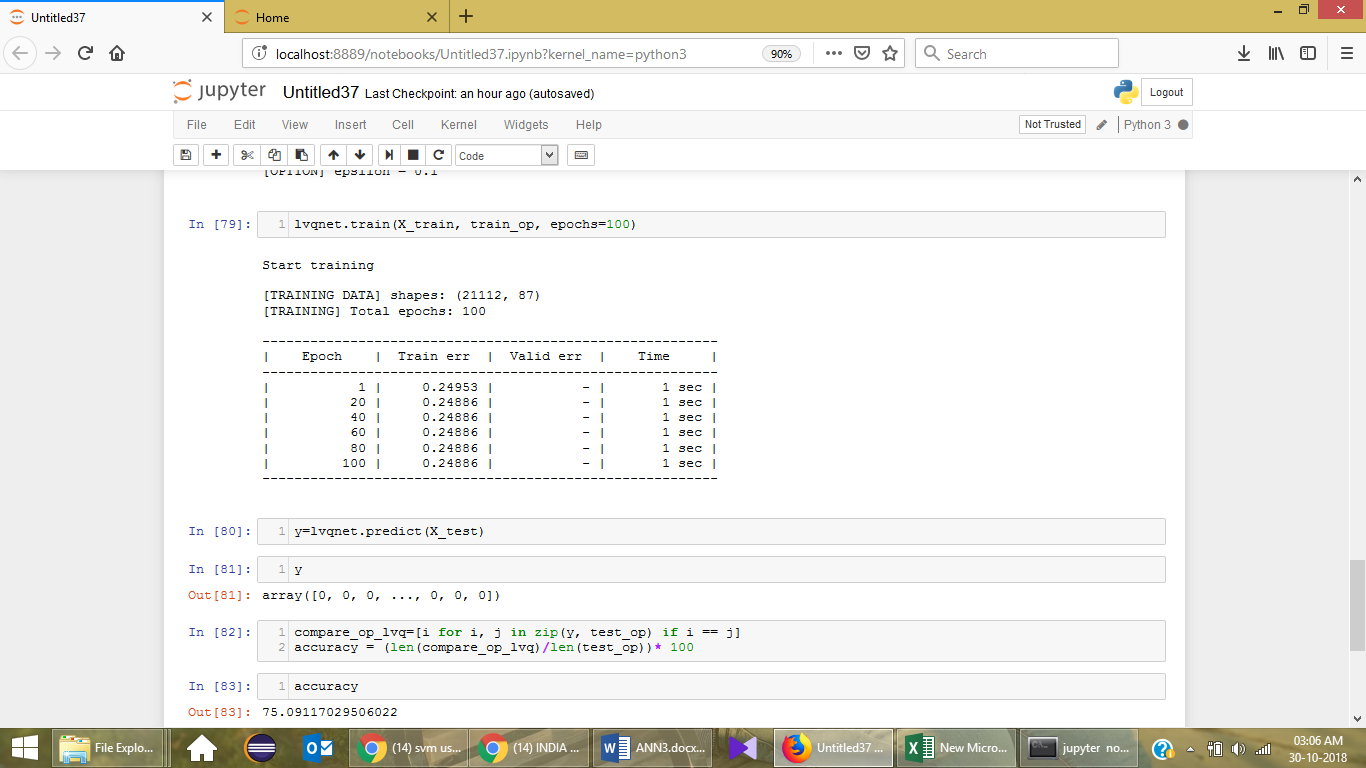
The plot between the accuracy and the value is shown below:

LVQ2:

In this part, the step function represents the learning rate for LVQ. The performance of LVQ2 with respect to this parameter is shown below:

|  |  |
| --- | --- |
| Step | LVQ2 |
| 1 | 0.75 |
| 0.5 | 0.7509 |
| 0.25 | 0.75 |
| 0.01 | 0.79 |
| 0.001 | 0.81 |

As the step function value is decreased (x axis), the accuracy value of the model gets high as shown.



**SVM**:

**Implementation:**

Support vector machines classifiers form a separating hyperplane in the high dimensional data space so that it can easily classify the labels for the data objects or instances. SVM is applied on the given data with the package used from sklearn.

As the gamma value (kernel coefficient) is being reduced, the accuracy value of the SVM classifier got increased as shown below:

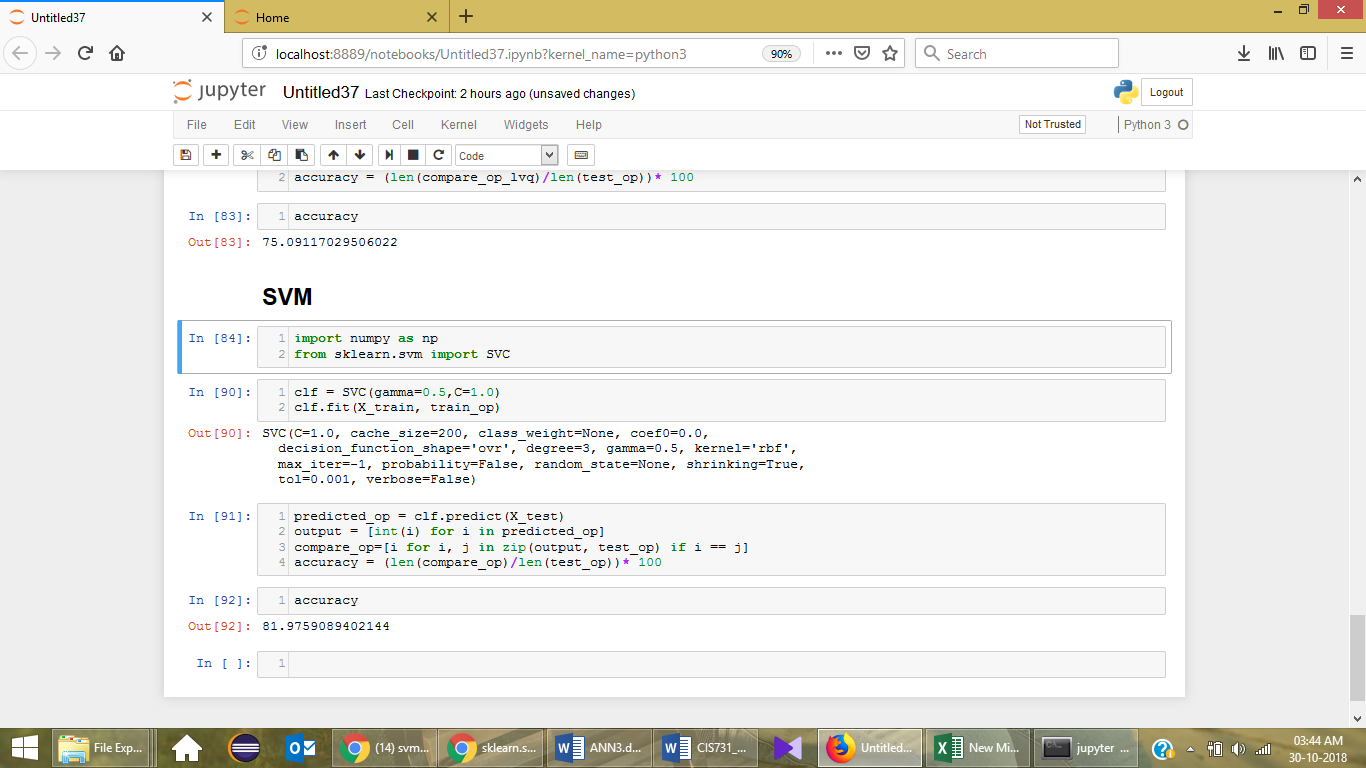
|  |  |
| --- | --- |
| **Gamma (rbf)** | **SVM** |
| 1 | 0.78 |
| 0.5 | 0.81 |
| 0.25 | 0.82 |
| 0.01 | 0.84 |
| 0.001 | 0.84 |

The performance of SVM based on different kernel is also shown below:

|  |  |
| --- | --- |
| **Kernel** | **SVM** |
| Linear | 0.849 |
| poly | 0.822 |
| rbf | 0.84 |
| sigmoid | 0.832 |

Kernel Coefficient value vs Accuracy

Types of Kernel vs Accuracy:



These are some of the algorithm parameters that alter the accuracy of the algorithm. Few more parameters are also there like the number of iterations, etc can contribute to the overall accuracy of the model as well.