

Read Me

MULTINOMINAL NAIVE BAYES CLASSIFIER

The final project is implemented in various programming languages.

The project contains implementations for the classifiers below:

1. Multinomial Naive Bayes Classifier (Python and Java)
2. Multiclass Logistic Regression (Python)
3. Multi Class perceptron (Java)
4. Neural Networks (Octave)

Installation of Packages and Prog.Languages:

The Python 3.3 version is used to implement the project. Hence, either python 3.3 or higher version is needed to execute the program.

Certain free/open source packages were used to ease the process of input training and test set.

They are:

Pandas

NumPy

Steps to install the Install the packages:

The setup.py file is attached with this project.rar file. Double-click the file and it installs certain tools in python directory.

After that, in command prompt, change the directory to C:\Python33\Tools\Scripts and enter "easy_install.exe numpy". This will install numpy package on the system

```
C:\Windows\system32\cmd.exe

C:\>Python27\Scripts\easy_install.exe numpy
Searching for numpy
Reading http://pypi.python.org/simple/numpy/
Reading http://numpy.scipy.org
Reading http://sourceforge.net/project/showfiles.php?group_id=1369&package_id=175103
Reading http://numeric.scipy.org
Best match: numpy 1.6.1
Downloading http://pypi.python.org/packages/2.7/n/numpy/numpy-1.6.1.win32-py2.7.exe#md5=30bec16292be262bd78ff1878a7d8953
Processing numpy-1.6.1.win32-py2.7.exe
numpy.__import_tools: module references __file__
numpy.__import_tools: module references __path__
numpy.core.generate_numpy_api: module references __file__
numpy.core.scons_support: module references __file__
numpy.core.setup: module references __file__
numpy.core.setup.common: module references __file__
numpy.core.tests.test_records: module references __file__
numpy.core.tests.test_regression: module references __file__
numpy.distutils.exec_command: module references __file__
numpy.distutils.misc_util: module references __file__
numpy.distutils.npy_pkg_config: module references __file__
numpy.distutils.system_info: module references __file__
numpy.distutils.command.build_src: module references __file__
```

After that, enter "easy_install.exe pandas" to install pandas.

```
C:\Windows\system32\cmd.exe

C:\>Python27\Scripts\easy_install.exe pandas
Searching for pandas
Reading http://pypi.python.org/simple/pandas/
Reading http://pandas.pydata.org
Reading http://pandas.sourceforge.net
Best match: pandas 0.7.3
Downloading http://pypi.python.org/packages/2.7/p/pandas/pandas-0.7.3.win32-py2.7.exe#md5=e1f7eb58eaf7d5b2c37d29c827599168
Processing pandas-0.7.3.win32-py2.7.exe
creating 'c:\users\saveenr\appdata\local\temp\easy_install-4iinwv\pandas-0.7.3-py2.7-win32.egg' and adding 'c:\users\saveenr\appdata\local\temp\easy_install-4iinwv\pandas-0.7.3-py2.7-win32.egg.tmp' to it
creating c:\python27\lib\site-packages\pandas-0.7.3-py2.7-win32.egg
Extracting pandas-0.7.3-py2.7-win32.egg to c:\python27\lib\site-packages
Adding pandas 0.7.3 to easy-install.pth file

Installed c:\python27\lib\site-packages\pandas-0.7.3-py2.7-win32.egg
Processing dependencies for pandas
Searching for python-dateutil<2
Reading http://pypi.python.org/simple/python-dateutil/
Reading http://labix.org/python-dateutil
Best match: python-dateutil 1.5
Downloading http://labix.org/download/python-dateutil/python-dateutil-1.5.tar.gz

Processing python-dateutil-1.5.tar.gz
Running python-dateutil-1.5\setup.py -q bdist_egg --dist-dir c:\users\saveenr\appdata\local\temp\easy_install-sdwxda\python-dateutil-1.5\egg-dist-tmp-xfpcj9
Removing python-dateutil 2.1 from easy-install.pth file
Adding python-dateutil 1.5 to easy-install.pth file

Installed c:\python27\lib\site-packages\python_dateutil-1.5-py2.7.egg
Finished processing dependencies for pandas

C:\>
```

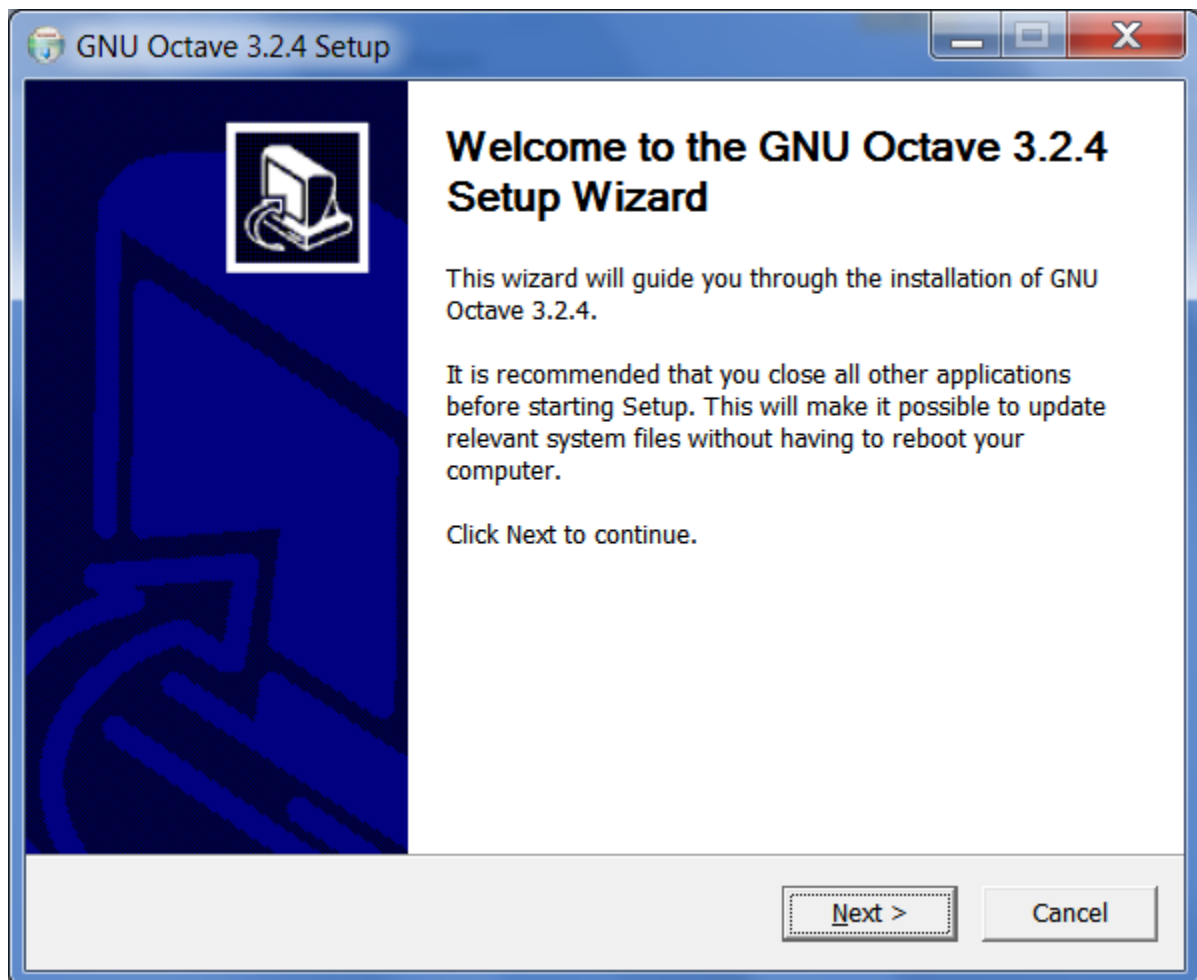
Octave Installation:

Neural Network is implemented in Octave for easy calculation of Cost function, Delta and Convergence of weight vector. The inbuilt function **fmincg** is called for minimizing the cost function. The dataset are transformed into a data matrix and injected as input units in neural networks.

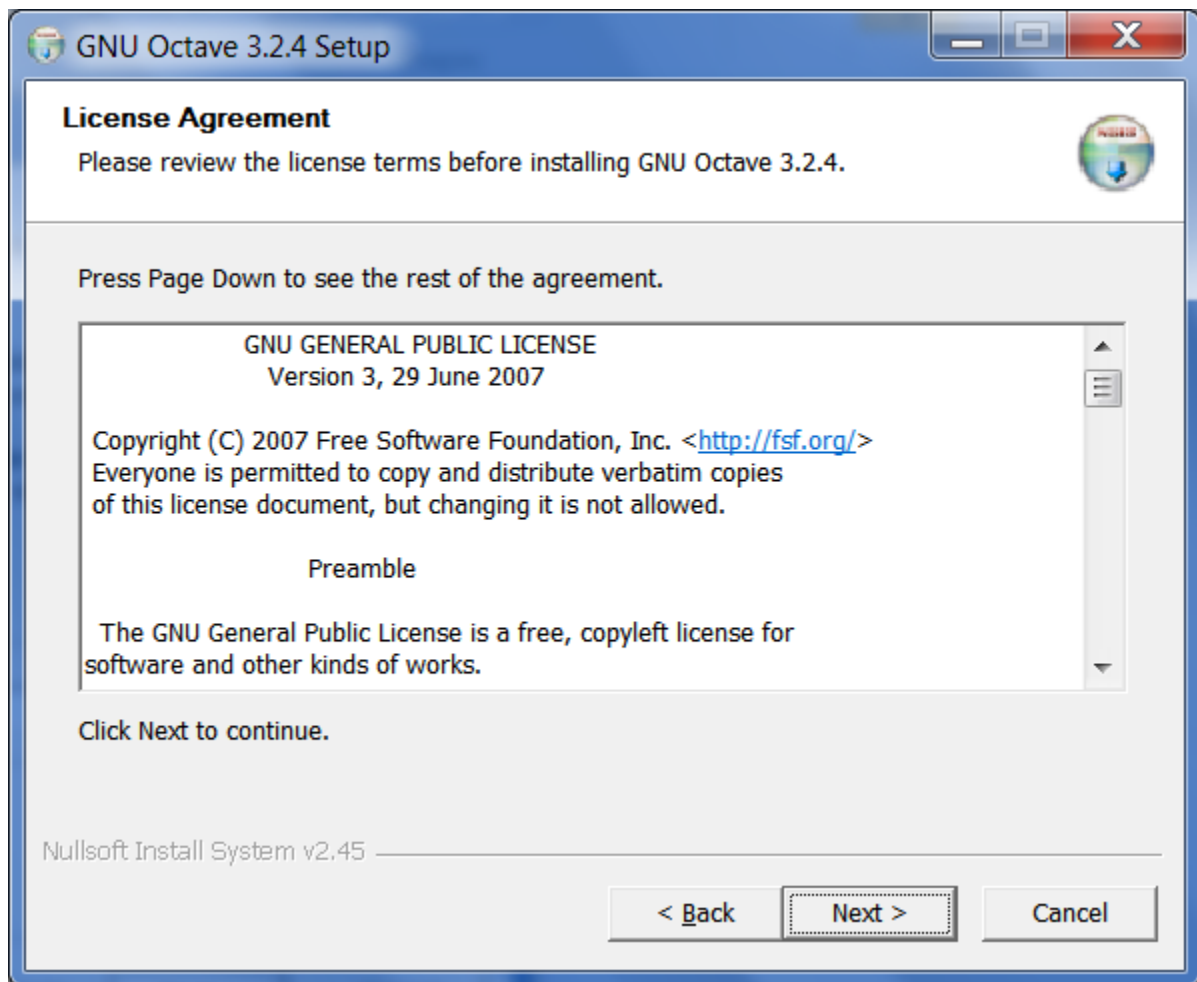
Installation of Octave software:

Windows Instructions

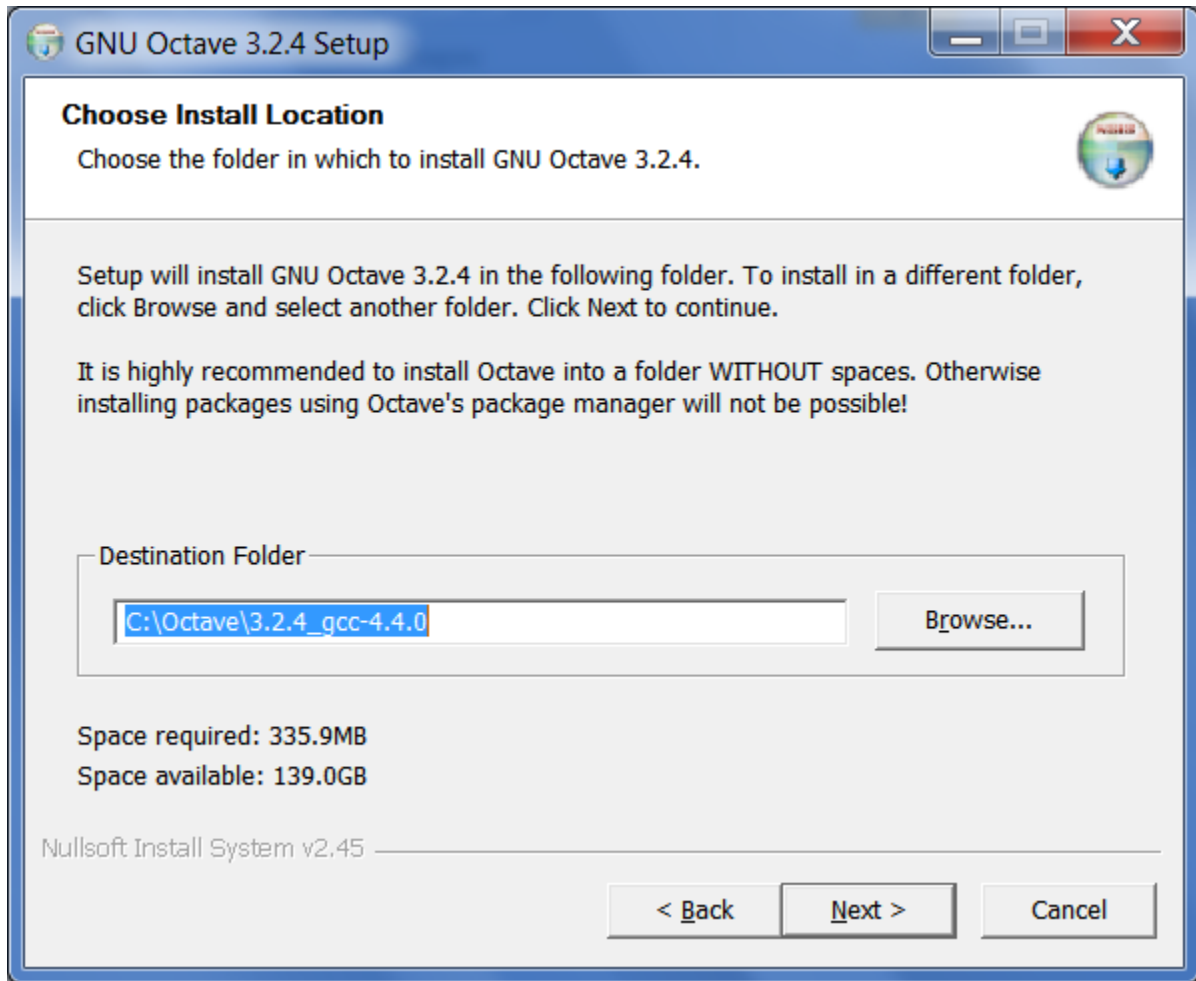
1. **Download** Octave: [Octave-3.2.4_i686-pc-mingw32_gcc-4.4.0_setup.exe](#)
2. **Run** the file and when prompted to allow the program to make changes to this computer, click **Yes**.
3. You should see the following installation screen. Click **next** to continue.



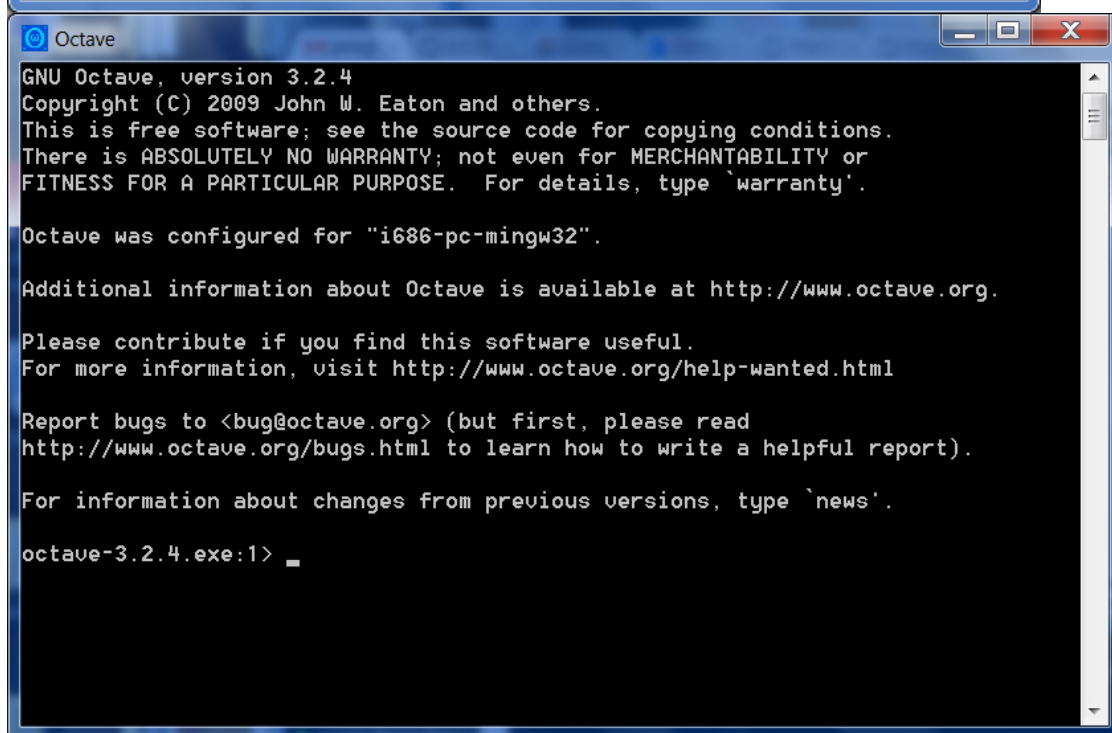
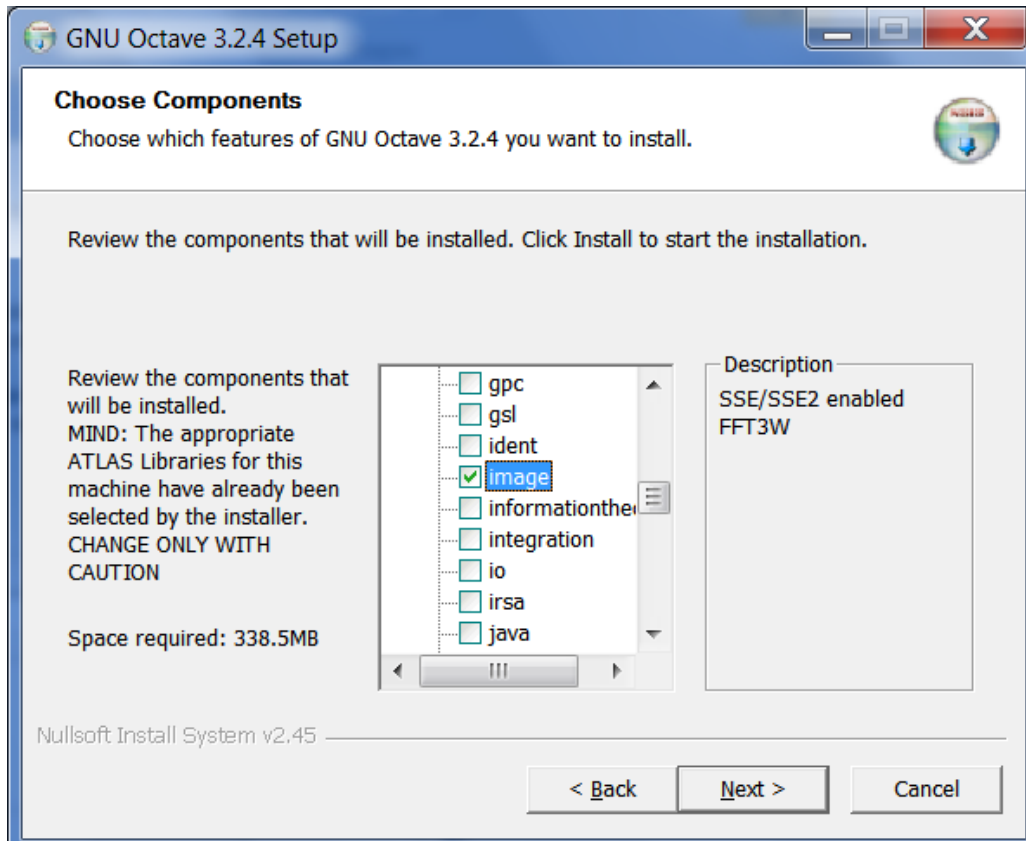
4. The next screen shows the license agreement. Click **next** to continue.



5. In the following screen, you can choose where to install Octave to. The default installation directory is recommended. Click **next** to continue.



6. Next, you can choose which components to install. In this class, we will often use functions from the **image** package. **Scroll** down the list and **select** the image package to be installed. Then, click **next** to continue. Then, Click **Finish** to complete the installation.



Functionalities of Naive Bayes Classifier:

Train_read_files : Read the training dataset and convert it to into a Data Matrix using Numpy and Pandas.

Naive Bayes : Functionality where probability of the class for give record is calculated

Accuracy : The accuracy is calculated.

Execution Time: 3 to 4 mins

Functionalities of Logisitic Regression:

Train_read_files : Read the training dataset and convert it to into a Data Matrix using Numpy and Pandas.

Predict LR : Test file is processed and Cost function is calculated and the accuracy is calculated.

Execution Time: 30 to mins (Hard iterations 500 for each class)

Functionalities of Neural Network:

neural_networks.m - Main File where initialization of input layer, hidden layer and output layers are done.

nnCostFunction.m - Cost Function and Delta calculation. Both Forward Propagation and Back Propagation is implemented.

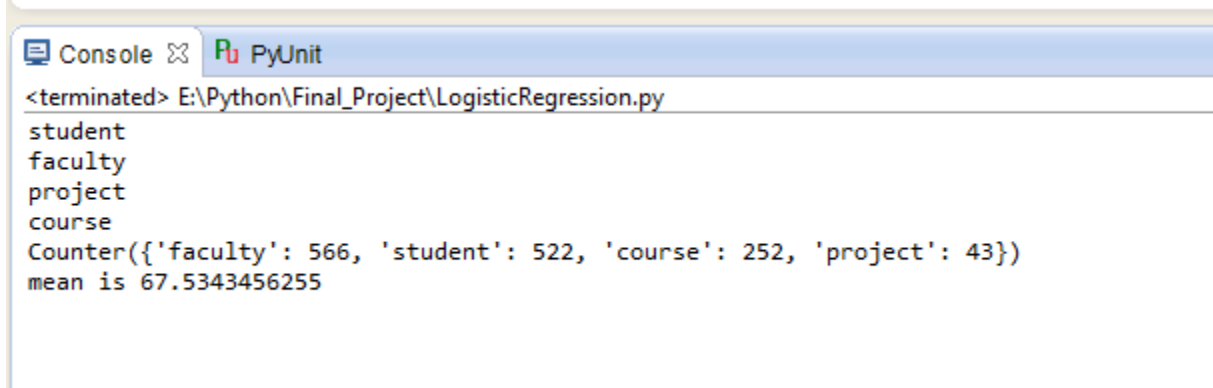
predict.m - Predict the classes of the test set.

randInitializeWeights.m - Random values initialized to the weight vectors for input layer and hidden layer.

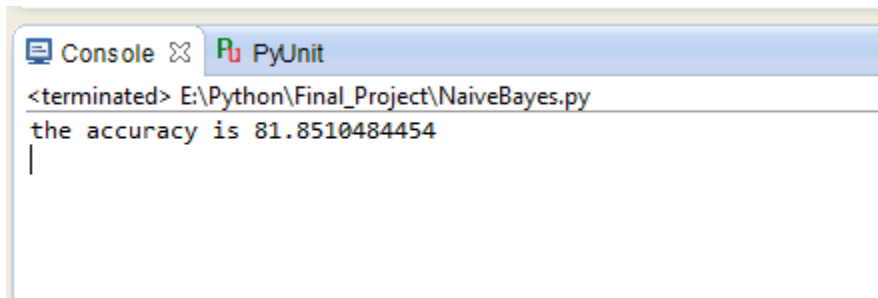
sigmoidGradient.m - sigmoid calculation for training data set.

Execution Time: More than 40 mins (1 hard iteration for converegence)

Output screen shots of Each Classifiers:



```
<terminated> E:\Python\Final_Project\LogisticRegression.py
student
faculty
project
course
Counter({'faculty': 566, 'student': 522, 'course': 252, 'project': 43})
mean is 67.5343456255
```

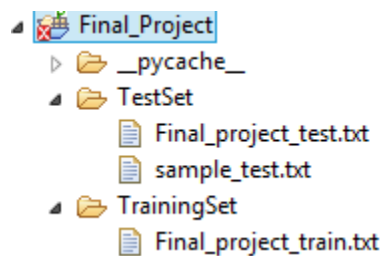


```
<terminated> E:\Python\Final_Project\NaiveBayes.py
the accuracy is 81.8510484454
|
```

Input Files:

Final_project_train.txt - Training File should be placed in TrainingSet Folder.

Final_project_test.txt - Test File should be placed in TestSet Folder.



train_file.txt - Training File for Octave (70 mb)

train_y.txt - Class Values of training dataset

test_file.txt - Test file for Octave(34.2 mb)

test_y.txt - Class values of Test dataset

All the files should be placed in same octave implementation folder.

(DropBox Link:

<https://www.dropbox.com/sh/nv5etpz8oj6z0am/vU0VymPPiv>)

```
Octave-3.2.4
Loading Data ...
ans =
    2785    6483
ans =
    2785         1
Initializing Neural Network Parameters ...
Cost Function J(theta): 2.772589
Training Neural Network...
Iteration    1 : Cost: 2.195222e+000
```

```
Octave-3.2.4
Initializing Neural Network Parameters ...
Cost Function J(theta): 2.772589
Training Neural Network...
Iteration    10 : Cost: 2.129200e+000
Visualizing Neural Network...
Program paused. Press enter to continue.
ans =
    1383    6483
ans =
    1383         1
Training Set Accuracy: 39.045553
octave-3.2.4.exe:4> _
```

ReadMe FOR JAVA IMPLEMENTATIONS

This readme section consists of 2 sub-sections, describing instructions to run and verify the **JAVA implementations** of

- **Multinomial Naïve Bayes Classification**
- **MultiClass Perceptron Classification**

algorithms.

1. Multinomial Naïve Bayes Classification Implementation

The jar files are present in the Jar Executables Folder along with the dataset folders.

Command used to run the jar file

Format: C:\Users\....ML_Project\Jar Executables>java -jar MultinomialNaiveBayesClassifier.jar

Run Time: ~1 Sec.

Output Format:

*******Train File Count : Course : 620 Faculty : 745 Project : 335 Student : 1085**

*******Whole vocab Set size : 6483**

Prior Probabilities C : 0.22262118491921004 F : 0.26750448833034113 P : 0.12028725314183124 S : 0.3895870736086176

*******MULTINOMIAL NAIVE BAYES TEXT CLASSIFICATION** *****

**TestCourseFiles : 279 TestFacultyFiles : 305
TestProjectFiles : 117 TestStudentFiles : 471**

**ClassifiedCourseFiles : 297 ClassifiedFacultyFiles : 393
ClassifiedProjectFiles : 158 ClassifiedStudentFiles : 548**

**Number of correctly classified Files : 1172
Number of wrongly classified Files : 224**

******Accuracy Percentage**** : 83.95415472779369 %**
Output is written to output.txt file

2. MultiClass Perceptron Classification Implementation.

The jar files are present in the Jar Executables Folder along with the dataset folders.

Command used to run the jar file

Format: C:\Users\....ML_Project\Jar Executables>java -jar
MultiClassPerceptronClassifier.jar

Run Time: ~3 Minutes. The number of iterations completed at any time is printed on the console. The limit on the number of iterations is 100.

Output Format:

Correct: 1121, Wrong: 275

******Accuracy in classifying the files****:**
80.30085959885386 %

Output is written to output.txt file