Online lab assignment “Classification”

# Pre-requisite to carrying out the assignment:

1. Go through and watch all the lab tutorials of modules 1&2:
2. Load from Sklearn the MINST dataset, using fetch\_openml.
3. Go through and watch all the lab tutorials related to modules 1,2 to understand how the code works.

# Assignment due date: end of week # 5

**General Instructions:**

Be sure to read the following general instructions carefully:

1. This assignment must be completed individually by all the students.
2. You will accompany your solution submission with an analysis report that contains your findings and the required screenshots.

# You will have to provide a demonstration video for your solution and upload the video together with the solution on eCentennial through the assignment link. See the video recording instructions at the end of this document.

**Assignment - exercise:** (100 marks)

Load & check the data:

1. Load the MINST data into a pandas dataframe named MINST\_firstname where first name is you name.
2. List the keys
3. Assign the data to a ndarray named X\_firstname where firstname is your first name.
4. Assign the target to a variable named y\_firstname where firstname is your first name.
5. Print the types of X\_firstname and y\_firstname.
6. Print the shape of X\_firstname and y\_firstname.
7. Create three variables named: some\_digit1, some\_digit2, some\_digit3. Store in these variables the values from X\_firstname indexed 7,5,0 in order.
8. Use imshow method to plot the values of the three variables you defined in the above point. Note the values in your written response.

Pre-process the data

1. Change the type of y to unit8
2. The current target values range from 0 to 9 i.e. 10 classes. Transform the target variable to 3 classes as follows:
   1. Any digit between 0 and 3 inclusive should be assigned a target value of 0
   2. Any digit between 4 and 6 inclusive should be assigned a target value of 1
   3. Any digit between 7 and 9 inclusive should be assigned a target value of 9 (Hint: you can use numpy.where to carry out the transformation on the target.)
3. Print the frequencies of each of the three target classes and note it in your written report in addition provide a screenshot.
4. Split your data into train test. Assign the first 60,000 records for training and the last 10,000

records for testing. (Hint you don’t need sklearn train test as the data is already randomized).

# Build Classification Models

Naïve Bayes

1. Train a Naive Bayes classifier using the training data. Name the classifier NB\_clf\_firstname.
2. Use the classifier to predict the three variables you defined in point 7 above. Note the results in your written response and compare against the actual results.

Demonstration Video Recording

Please record a short video (max 4-5 minutes) to explain/demonstrate your assignment solution. You may use the Windows 10 Game bar to do the recording:

1. Press the Windows key + G at the same time to open the Game Bar dialog.

2. Check the "Yes, this is a game" checkbox to load the Game Bar.

3. Click on the Start Recording button (or Win + Alt + R) to begin capturing the video.

4. Stop the recording by clicking on the red recording bar that will be on the top right of the program window.

(If it disappears on you, press Win + G again to bring the Game Bar back.)

You'll find your recorded video (MP4 file), under the Videos folder in a subfolder called Captures.

Submit the video together with your solution and written response.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Evaluation criteria | **Not acceptable** | **Below**  **Average** | **Average** | **Competent** | **Excellent** |
|  | **0% - 24%** | **25%-49%** | **50-69%** | **70%-83%** | **84%-100%** |
| Data exploration Visualization &  Pre-processing code  30% | Missing all requirements required | Some requirements are implemented. | Majority of requirements are implemented but some are malfunctioning. | Majority of requirements implemented. | All requirements are implemented  Correctly. |
| Model building Validation  &Testing  30% | No evidence of testing and evaluation of the requirements. | Minor evaluation and testing efforts. | Some of the requirements have been tested & evaluated. | Majority of requirements are tested & evaluated. | Realistic evaluation and testing, comparing the solution to the requirements. |
| Code Documentation  5% | No comments explaining code. | Minor comments are implemented. | Some code is correctly commented. | Majority of code is correctly commented. | All code is correctly commented. |
| Written analysis  Content  10% | Missed all the key ideas; very shallow. | Shows some thinking and reasoning but most ideas are underdeveloped. | Indicates thinking and reasoning applied with original thought on a few ideas. | Indicates original thinking and develops ideas with sufficient and firm evidence. | Indicates synthesis of ideas, in-depth analysis and evidences original thought and support for the topic. |
| Written analysis Format and organization  5% | Writing lacks logical organization. It shows no coherence and ideas lack unity. Serious errors. No transitions.  Format is very messy. | Writing lacks logical organization. It shows some coherence but ideas lack unity. Serious errors.  Format needs attention, some major errors. | Writing is coherent and logically organized. Some points remain misplaced.  Format is neat but has some assembly errors. | Writing is coherent and logically organized with transitions used between ideas and paragraphs to create coherence. Overall unity of ideas is present. Format is neat and correctly assembled. | Writing shows high degree of attention to logic and reasoning of all points. Unity clearly leads the reader to the conclusion.  Format is neat and correctly assembled with professional look. |
| Demonstration Video  20% | Very weak no mention of the code changes. Execution of code not demonstrated. | Some parts of the code changes presented.  Execution of code partially demonstrated. | All code changes presented but without explanation why. Code demonstrated. | All code changes presented with explanation, exceeding time limit. Code demonstrated. | A comprehensive view of all code changes presented with explanation, within time limit. Code demonstrated. |

1. Use 3-fold cross validation against the train data and note the results in your written response.
2. Use the model to score the accuracy against the test data, note the result in your written response.
3. Generate the accuracy matrix. Logistic regression
4. Train a Logistic regression classifier using the same training data. Name the classifier LR\_clf\_firstname. (Note this is a multi-class problem make sure to check all the parameters and set multi\_class='multinomial'). Set max\_iter to 1000 and tolerance to 0.1 in both cases.

Try training the classifier using two solvers first “lbfgs” then “Saga”. Make sure you note the results in both cases in your written response, and note the main differences in your written response with a written explanation.

1. Use the classifier that worked best from the above point to predict the three variables you defined in point 7 above. Note the results in your written response and compare against the actual results.
2. Use 3-fold cross validation against the training data and note the results in your written response.
3. Use the model to score the accuracy against the test data, note the result in your written response.
4. Generate the accuracy matrix
5. Generate the precision and recall of the model and note them in your written response.

Finally, in your written response compare the results from both models i.e. (The Naïve Bayes and the Logistic regression) and write your conclusions.