

COMP 237 - Online lab assignment “Logistic Regression”

Purpose:

The purpose of this Lab assignment is to:

1. To get hands-on experience of applying supervised machine Learning namely the Logistic regression algorithm to solve a business problem.
2. To get hands-on experience with data exploration and pre-processing for machine learning problems.
3. To get hands-on experience in data normalization.
4. To experiment with different train-test splits.

General Instructions:

Be sure to read the following general instructions carefully:

1. This assignment must be completed individually by all the students.
2. Only provide the requested screenshots and make sure to have a complete screenshot, partial screenshots will not earn any marks.
3. 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.
4. In your 5-minute demonstration video you should explain your solution clearly, going over the main code blocks and the purpose of each module/class/method also demoing the execution of exercises #1 & 2. Youtube links and links to google drive or any other media are not acceptable, the actual recording must be submitted.
5. Any submission without an accompanying video will lose 70% of the grade.
6. In your analysis report make sure you provide an introduction and clearly state the facts and findings. Any submission missing Analysis report will lose 70%.

Submission:

There are three documents to be submitted for this assignment in one zipped folder (All subject to grading as per the rubric for this assignment):

1. For each exercise that requires code, please create a project folder and include all project python scripts/modules and screenshot of output, as needed. Name all python scripts your `firstname_linear.py`. Name the folder “Exercise#X_firstname”, where X is the exercise number and firstname is your first name. (In total **1** folders for this assignment).
2. For all questions that require written or graphic response create one “Word document” and indicate the exercise number and then state your response. Name the document “Written_responses_firstname”, where firstname is your firstname. (In total one word or pdf document).
3. All submissions need to be accompanied with a recorded demonstration video not to exceed 5 minutes in length, focus on showing the key code functionalities and run the code.

Create one zipped folder containing all of the above, name it Logistic_firstname where firstname is your firstname.

Pre-requisite to carrying out the assignment:

1. Download from the course shell the following comma separated file: studentsdropout.csv. This file contains the details of each student enrolled in different undergraduate degrees, and also their academic success.

A brief description of the column names of the dataset is, as follows:

Field
Marital status
Course
Daytime/evening attendance
Admission grade
Displaced
Educational special needs
Tuition fees up to date
Gender
Scholarship holder
Age at enrollment
International
Curricular units 1st sem (grade),
Curricular units 2nd sem (grade),
Academic_success – one of two categories (dropout, and graduate) at the end of the normal duration of the course

2. Go through and watch all “Linear & Logistic” lecture and lab tutorials related to modules # 5 & 6 to understand the concepts and the presented code.

Assignment - exercises:

1. **Exercise # 1:** students dropout analysis using logistic regression (100 marks)

Requirements:

- a. Get the data :

1. Load the “studentsdropout.csv” data into a data frame, name the dataframe studentdropout_firstname , where *firstname* is your *firstname*.

- b. Initial Exploration:

1. Display (print) the first 5 records.
2. Display (print) the names, types and counts (showing missing values per column). Use pandas built in method [info](#). For more info checkout: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.info.html>
3. Display (print) the shape of the dataframe.

4. Display (print the unique values for the following columns : ("Educational special needs", "Displaced") c. Data visualization
1. Use pandas crosstab and matplotlib to generate the following diagrams plots:
 - a. A bar chart showing the Gender versus the # of Academic_success class. Give an appropriate name for the x and y axis in addition to an appropriate title that includes your name.
 - b. A bar chart showing the # of 'Tuition fees up to date' versus the # of Academic_success. Give an appropriate name for the x and y axis in addition to an appropriate title that includes your name.
 - c. Analyze both plots and write a conclusion from each plot in your written response.
2. Use pandas scatter matrix to plot the relationships between the number of graduates and the all features (attributes). Analyze the output and write some conclusions in your written response. For more info checkout:
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.plotting.scatter_matrix.html

d. Data transformation (round #1):

1. Drop the four columns you identified in point (b.4) above.
2. Using "Get dummies" transform all the categorical variables in your dataframe into numeric values. (There should be two columns)
3. Attach the newly created variables to your dataframe and drop the original columns.
4. Remove the original categorical variables columns. Use pandas drop method and select the correct argument values. For more info checkout :
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.drop.html>
5. Replace the missing values in the 'Age at enrollment' with the mean of the age.
5. Change all column types into float.
6. Write a function that accepts a dataframe as an argument and normalizes all the data points in the dataframe. Use pandas .min() and .max().
 Below the formula for normalization:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Call the new function and pass as an argument your transformed dataframe. By now all your data is numeric.

7. Display (print) the first two records.
8. Use pandas.hist to generate a plot showing all the variables histograms. Set the figure size to 9 inches by 10 inches. For more info, checkout :

<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.hist.html>

9. Create histograms for “Curricular units 1st sem” & “Curricular units 2nd sem” and write in your written response some highlights (Conclusions).
10. Split the features into a dataframe named `x_firstname` and the target class into another dataframe named `y_firstname`.
11. Using Sklearn “`train_test_split`” split your data into 70% for training and 30% for testing, set the random seed to be the last two digits of your student ID number.
Store the training data in a dataframe named: `x_train_firstname` for the features (predictors) and the training labels `y_train_firstname`. Store the test data as follows: `x_test_firstname` and `y_test_firstname`.

e. Build & validate the model

1. Using sklearn fit a logistic regression model to the training data. Name the model `firstname_model`.
2. Display (print) the coefficients (i.e. the weights of the model). Since we have multiple feature you can use pandas, zip and np transpose, something as follows to get a neat table:

```
pd.DataFrame(zip(x_train_mayy.columns, np.transpose(model_mayy.coef_)))
```

3. Cross validation:

1. Use Sklearn `cross_val_score` to validate the model on the training data.
2. Set the number of folds cv to 10.
3. Repeat the validation for different splits of the train/ test. Start at test size 10% and reach test size 50% increasing your test sample by 5%.
4. In each run print out the minimum, mean and maximum accuracy of the score.
5. Note these results in your writer report and recommended the best split scenario.

(Hint: you will need a loop something like this: for i in np.arange (0.10, 0.5, 0.05):)

b. Test the model

1. Rebuild the model using the **70% - 30%** train/test split.
2. Define a new variable `y_pred_firstname` where `firstname` is your firstname, store the predicted probabilities of the model in this variable (*hint: use `predict_proba`*)
3. Define another variable name it `y_pred_firstname_flag`, store in the `y_pred_firstname` after transforming the probabilities into a boolean value of true or false based on a threshold value of **0.5**. (*hint : `y_pred_mayy_flag = y_pred[:,1] > 0.5`*)
4. From sklearn metrics import : `confusion_matrix`, `accuracy_score`, `classification_report`

5. Print out the accuracy of the model on the test data.
6. Print out the confusion matrix.
7. Print out the classification report.
8. Write down and note the values of : accuracy, precision and re-call. Compare the precision and call, and add your observation to the Analysis report.
9. Create a plot of the ROC Curve. Add your observation to the Analysis report.
10. Note the values of Specificity and Sensitivity and compare them.
11. Repeat steps 3 to 6 with changing the threshold value to **0.60**
12. Compare the accuracy on the test data with the accuracy generated using the training data.
13. Compare the values of accuracy, precision and re-call generated at the threshold **0.5** and **0.75** .

----- End of Exercises -----

Rubric

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.

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.

Or you can use any other video recording package freely available.