COMP237 – Logistic Regression Lab

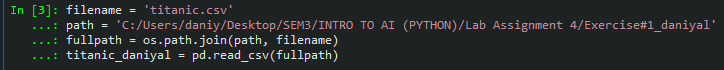
Daniyal Ahmed – 301152472

**Exercise 1 Linear Regression with titanic.csv**

First we import the required modulesText

Description automatically generated

Then I created a dataframe named titanic\_daniyal containing the data from titanic.csv



Then did some initial exploration:

Text

Description automatically generated

Text

Description automatically generated

A picture containing text, plaque

Description automatically generated

Identify 4 columns that are not going to be useful for the model:

* PassengerID – It is just indexing from 1-891, not required as 0-890 indexing with dataframes already available
* Ticket – Unique values with no relevance
* Name – Unique values and analysis doesn’t require names of subjects
* Cabin – Many missing values

Displayed the unique values for “Sex” and “Pclass”

Text

Description automatically generatedText

Description automatically generated

Using matplotlib and crosstab generate a bar chart showing # of survived versus passenger class with names for x and y axis and title with name

* It can be seen that many in the 3rd passenger class did not survive, while approximately equal amounts of class 2 and 1 passengers survived and died.
* Passenger class 3 might be its economy class and didn’t have safety measures as good as others, causing most of its members to die

Chart

Description automatically generated

Using matplotlib and crosstab generate a bar chart showing # of survived versus gender with names for x and y axis and title with name

* In this plot it can be seen that in terms of males, more died than survived
* In terms of females however, more females actually survived than the amount that didn’t.Chart, bar chart

  Description automatically generated

Plot a scatter matrix between Gender, Passenger class, Fare, Number of siblings/spouses abroad, Number of parents/children abroad.(See next two pages)

Graphical user interface, text

Description automatically generatedChart

Description automatically generated

Conclusions:

* The most expensive passenger class is class 1, and the least expensive is class 3
* Most people with 0 or 1 sibling/spouse pay less than 100 in fare
* Many people with 0, 1, and 2 parents/children abroad pay 100 or less for their fare

Drop the 4 columns identified earlier:

Text

Description automatically generated

Using get\_dummies convert all categorical variables in your dataframe into numeric values(2 columns)

Text

Description automatically generated

Drop original categorical variables 

Replace missing values in age with mean of age:



Change all column types to float

Text

Description automatically generated

Create function to normalize data points in dataframe

Text

Description automatically generated

Call function passing titanic\_daniyal as an argument 

Display the first 2 records

Text

Description automatically generated

Create histogram with size 9 by 10 inches showing all variables

Text

Description automatically generated A picture containing text, crossword puzzle

Description automatically generated

In previous histogram it can be seen that:

* Most people embarked with S
* Fewer people chose C
* Fewest people chose Q

Split features into dataframe named x\_daniyal(features) and the target class into y\_daniyal(survived)

A screenshot of a computer

Description automatically generated with medium confidence

Set seed to 72 (last 2 digits of student number) 

Split model into test and training sets Text

Description automatically generated

Create a logistic regression model and fit it with the training sets

Text

Description automatically generated

Print out coefficients using pandas, zip and transpose

Text

Description automatically generated

Use cross\_val\_score to validate model on training data

A screenshot of a computer

Description automatically generated with medium confidence

Repeat validation for different splits of train/test from 10% to 50% increasing by 5% everytime. Print out minimum, mean and maximum accuracy of score. Text

Description automatically generated Text

Description automatically generated

From these results it can be seen that:

* the tests produce a minimum score of between 69-75
* the tests also produce a maximum between 81-87
* The means range between 76-80
* The best split scenario based on the mean accuracy would be the testing size of 35%

Next I rebuilt and fit the model using 70/30 train/test split Text

Description automatically generated

Then I defined y\_pred\_daniyal to be predicted probabilities of the model 

Then I defined a var named y\_pred\_daniyal\_flag that transforms y\_prob\_daniyal into boolean values based on threshold of 0.5 

Then I printed out the accuracy of the model on the test data, which can be seen to be 80.2%Text

Description automatically generated

Then printed out confusion matrix Text

Description automatically generated

Then printed out classification report, the precision can be seen to be 80% for both macro and weighted averages, the precision for true(1.0) is 76% and for false(0.0) is 83%, and the recall is 85% for 0 and 73% for 1. A screenshot of a computer

Description automatically generated with medium confidence

Then I repeated the previous steps with a threshold of 0.75

Calendar

Description automatically generated

Accuracy: 76.12%

Precision: 73% for 0 and 89% for 1

Recall: 96% for 0 and 46% for 1

* It can be seen that the accuracy with threshold of 0.5 is much more accurate than that with threshold of 0.75.
* The precision can be seen to be higher for the false values with threshold of 0.5, while it is higher for true values with threshold of 0.75
* The recall was higher for 0 in the threshold of 0.75, while it was SIGNIFICANTLY higher for 1 with a threshold of 0.5. This means that the model had a better ability finding all relevant cases in the data set for the true values in threshold of 0.5 and false values in threshold of 0.75