ML lab 5 helper

<https://www.youtube.com/watch?v=RgO8BBNGB8w&list=PLjy4p-07OYzulelvJ5KVaT2pDlxivl_BN>

Pairs:

Povetry – unemployed

Povetry – black

Povetry – Hispanic

Professional – income

Office – income

Income – income per cap

income per capita

Cross validation: <https://github.com/eclarson/MachineLearningNotebooks/blob/master/PDF_Slides/archive/ML_6_history_wide_deep.pdf>

Also contain wide and deep NN

Also cross validation:

<https://smu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=812b717f-4e64-4ed8-bed9-ab88014c6a72> at 19:13, start at 12:00

The final data set I am using contains 33 numerical attributes plus 5 categorical columns I made for the states, gender, occupation, race’ and income. I deleted the original columns of ID and county since they do not contribute any data. Also, I deleted the missing data. For the target set, I categorized the data into four groups as shown above. Finally, I normalize the numeric values in the data set on a scale of -1 to 1. I did not one hot encoded the categorical data since Keras will do it for me. After splitting the data into train and test sets, I will separate again the numeric values and the categorical columns to train the deep neural network.

When I splinted the target into four different grouped, I compare those points according to the average in the U.S. as well the average in third world countries. The difference between the first and the second groups (0 and 1) was made just in order to split it equally. Therefore, false positive or true negative between those two categories should not affect the evaluation my algorithm. However, the difference between the second to third group (1 and 2), from lower than the U.S average to higher than that and the difference between the third and the last group (2 and 3), moving from “watch” group to the extreme poverty group will be a heavy weight. With that being said, for those differences, predicting an area to be in a poorer group than it actually is, is less weighted than predicting an area to be in a “richer” group because it is better to be in a “safer mode” where you take more care even if it is not necessary, other than do not take care of an area that actually need it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Pred 0 | 1 | 2 | 3 |
| Real0 | 0 | 1 | 4 | 8 |
| 1 | 1 | 0 | 3 | 5 |
| 2 | 8 | 6 | 0 | 3 |
| 3 | 10 | 8 | 6 | 0 |

Since the data contains more than 72K different elements, defined by 34 attributes, it is large enough dataset to have variation in the train set as well as in the test one. if we shuffle it and choose our 20% test set randomly, it will be most likely the same result as for any other general case.

ss = StandardScaler()

df\_train[col] = ss.fit\_transform(df\_train[col].values.reshape(-1, 1))

this is for normalize.

From the 3.26 video:

How make make a MLP in Keras

1:28:20 – dense layer is NN layer with units= the number on neurons (n\_hidden form lab 4)

I should use activation=’relu’

X = Dense(units=10,activation=’relu’)(input) // the last input is the prev layer.

All shown in Ln[8] in notebook 10.

Ln[10] – verbose=1 means give feedback

Yhat = np.round(yhat\_proba) // will give me the matrix with the results.

3.31 video:

For Ln[9] should see if there are more options for the optimizer as well as the loss

Add model.summary()

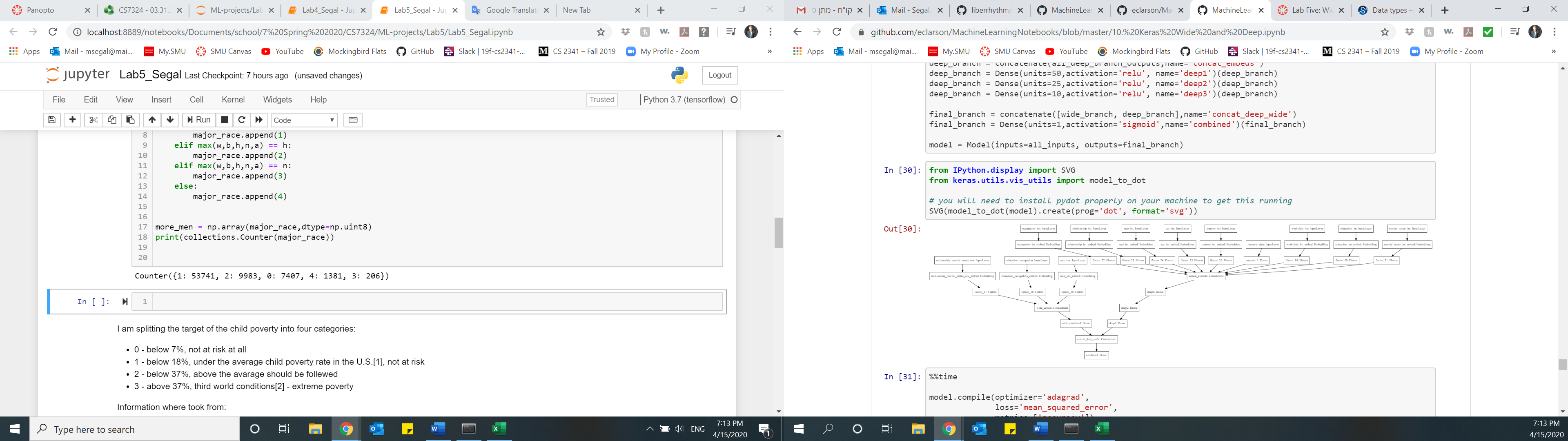
Better way for one hot encoded is to use the sklearn – OneHOtEncoder in ln{11]

1:12:00 – create training for each categorical set

I need to change the categorical data I made and also make more categorical data by using:

Major gender (0,1,2 – 1 for pretty even), most common race, income , most common work place,

Then I will need to make the crossed groups that I mentioned before. Also make triple group.



ROC curve

<https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html#sphx-glr-auto-examples-model-selection-plot-roc-py>

Out[34] for the ROC