

Computer Science Department
CS672 – Introduction to Deep Learning (CRN: 72900)
Fall 2022

Project #2 / Due 21-Nov-2022

Build a Deep Learning model (**based on Neural Networks**) that provides reliable and improved accuracy of a NYC Yellow Taxi **trip/ride duration**.

You used the NYC/TLC Yellow Cab dataset, in Project #1, to find correlation among the various variables in order to improve ride time predictions and predict the trip duration of a taxi ride taking into account the different factors that affect the ride duration. Along with the above-mentioned dataset, one more file should be included in your analysis, which involves the climatic conditions of the city.

Both of these datasets should be combined using pre-processing techniques to create a single dataset that can be used further for accurate trip duration prediction.

Get the **NYC climatic data** from the Meteostat portal.

Here is the url to download the Jan-2020 climate data (*):

<https://meteostat.net/en/place/3BBKPQ?t=2020-01-01/2020-01-31>

(Note, the temperatures are in Celsius degrees and not in Fahrenheit!)

NYC-Weather-Jan-2020 * x											
	A	B	C	D	E	F	G	H	I	J	K
1	date	tavg	tmin	tmax	prcp	snow	wdir	wspd	wpgt	pres	tsun
2	1/1/2020	3.6	1.7	5	0	0	264.6	17.3		1008.2	
3	1/2/2020	4.7	0.6	8.9	0	0	218.2	12.4		1013.9	
4	1/3/2020	7.6	6.7	8.3	2.8	0	235.5	8.4		1010.2	
5	1/4/2020	8.2	6.7	9.4	5.3	0	325.2	5.7		1003.7	
6	1/5/2020	4.6	2.8	7.2	0	0	300.1	8.2		1010.1	
7	1/6/2020	2.7	-0.5	7.8	0.8	30	239.2	16.6		1014.4	
8	1/7/2020	4.7	3.3	6.7	0	0	286	10.4		1016.2	
9	1/8/2020	2.6	-0.6	6.7	0	0	269.5	13.4		1016.6	
10	1/9/2020	-0.6	-3.3	2.2	0	0	272.3	9.1		1038.8	
11	1/10/2020	6.4	1.1	11.7	0	0	265.6	14.5		1034.5	
12	1/11/2020	14.2	8.9	21.1	0	0	192.7	19.8		1022.6	
13	1/12/2020	15.7	9.4	19.4	1	0	247.8	15.5		1018.1	
14	1/13/2020	6.1	1.7	7.8	0	0	13.4	10.4		1029	
15	1/14/2020	6.3	5	7.8	0.5	0	55	9.9		1025.3	
16	1/15/2020	8.2	6.1	11.7	0	0	269.4	9		1020.9	
17	1/16/2020	6.7	2.2	10	1.3	0	274.6	10.6		1015.1	
18	1/17/2020	-1.2	-3.9	2.2	0	0	321.2	8.8		1034.7	
19	1/18/2020	-2.2	-5	2.8	7.9	0	187.3	10.2		1028.5	
20	1/19/2020	4	0.6	7.2	0	0	269.1	10.2		1008.9	
21	1/20/2020	-1.7	-5	2.8	0	0	328.4	9.1		1024.2	
22	1/21/2020	-1.5	-4.4	3.3	0	0	323.7	8.9		1030.8	
23	1/22/2020	1	-2.2	5.6	0	0	286.8	8.1		1031.3	
24	1/23/2020	3.5	0	7.2	0	0	227.1	7.5		1029.4	

Perform **regression modeling analysis** on the ready-to-be-fed data into the following Neural Network based algorithms:

- 1) **MLP** (Multi-Layer Perceptron)
- 2) **Linear Regression** (TF/Keras Sequential model w/ no hidden layers)
- 3) **DNN** (Deep Neural Network with at least 2 hidden layers)

Split your dataset into training and validation datasets in 80%/20% ratio.
(Note: Since dataset is time-sensitive, be extra careful on splitting...)

At the **compiling** phase, use:

- **loss function:** Mean Square Error (MSE), Mean Absolute Error (MAE).
- **optimizer:** SGD, Adam, RMSProp.
 - Use various values for **learning rate (lr)**

After each **fit** (run each one for a single value of **epoch**=100), present/plot the training_loss vs validation_loss (could utilize TensorBoard GUI module)

Present the **best Regression Deep Learning model** (compare the above 3) and its corresponding parameters. Assume batch_size takes its default value of 32.

Using the best selected regression model, perform predictions by calling Keras **model.predict** module and review the loss.

Be aware of the following:

- A) Mean square error (MSE) and mean absolute error (MAE) are common loss functions used for regression problems. MAE is less sensitive to outliers.
- B) When numeric input data features have values with different ranges, each feature should be scaled independently to the same range.
- C) Overfitting is a common problem for DNN models

(*) While at the **Meteostat** portal, select '**location**' (New York, Wall Street) and '**date range**' (01/01/2020 to 01/31/2020).

New York / Wall Street ▼	⬇	⌘	📅 01/01/2020 - 01/31/2020
--------------------------	---	---	---------------------------

Note, the default 'date range' will be the current week...