# ITP 359, Spring 2025

# Homework 1 25 points Review of deep Neural Networks

# DUE: January 31, 2025, 11:59pm

**Predict New York Taxi fares using Deep Neural Networks (20 points)[[1]](#footnote-1)**

Predict the fare amount for a taxi ride in New York City given the pickup and dropoff locations. While you can get a basic estimate based on just the distance between the two points, this will result in an RMSE of $5-$8, depending on the model.

Download the data file of fares from Blackboard.

The file contains these columns.

* **key** - Unique string identifying each row. Comprised of **pickup\_datetime** plus a unique integer, but this doesn't matter, it should just be used as a unique ID field.
* **pickup\_datetime** - timestamp value indicating when the taxi ride started.
* **pickup\_longitude** - float for longitude coordinate of where the taxi ride started.
* **pickup\_latitude** - float for latitude coordinate of where the taxi ride started.
* **dropoff\_longitude** - float for longitude coordinate of where the taxi ride ended.
* **dropoff\_latitude** - float for latitude coordinate of where the taxi ride ended.
* **passenger\_count** - integer indicating the number of passengers in the taxi ride.

## Target

* **fare\_amount** - float dollar amount of the cost of the taxi ride.

For this homework, use tensorflow and keras to build and train a deep neural network to predict taxi fares. Here are the requirements.

1. Read the dataset into a dataframe. Parse the pickup\_datetime so that it is read as timestamp. (1)
2. Explore the dataset and determine what is the target variable. (1)

**fare\_amount**

1. Drop factor(s) that are not likely to be relevant for predicting the taxi fare*.* (2)
2. Extract the *weekday* and the *time* (hours/minutes) from the pickup\_datetime. Store in the data frame. Drop the pickup\_datetime. (2)
3. Compute the distance between pickup\_longitude and dropoff\_longitude by using this approximate formula for relatively close points in the earth (e.g. within NYC) (2)

*a = diff in longitude of two points \* 54.6*

*b = diff in latitude of two points \* 69.0*

*distance in miles = sqrt (a^2 + b^2)*

1. Drop *latitude* and *longitude* columns. (1)
2. Assign X (features variables) and y (target variable) (1)
3. Build a keras *sequential* model with two *dense* layers. Number of neurons is your choice (> 50). Activation function is your choice. (3)
4. Add a *dense* output layer. How many neurons in this layer? (2)
5. Compile the model with optimizer as *adam*, loss as *mean squared error*, metrics as *mean squared error.* (2)
6. Train the model. Partition with a split of 70/30. Epochs > 30. (2)
7. Display the *plot of the training and validation accuracy vs epoch. (2)*

A graph of training and validation

AI-generated content may be incorrect.

1. Print the *R-squared* score. (1)

**R-squared score: 0.762594622429551**

1. Finally, print the prediction of taxi fare for 2 passengers riding 3.2 miles at 3:20 pm on a Friday. (3)

**Predicted taxi fare for 2 passengers riding 3.2 miles at 3:20 PM on a Friday: 16.43**

1. https://www.kaggle.com/competitions/new-york-city-taxi-fare-prediction [↑](#footnote-ref-1)