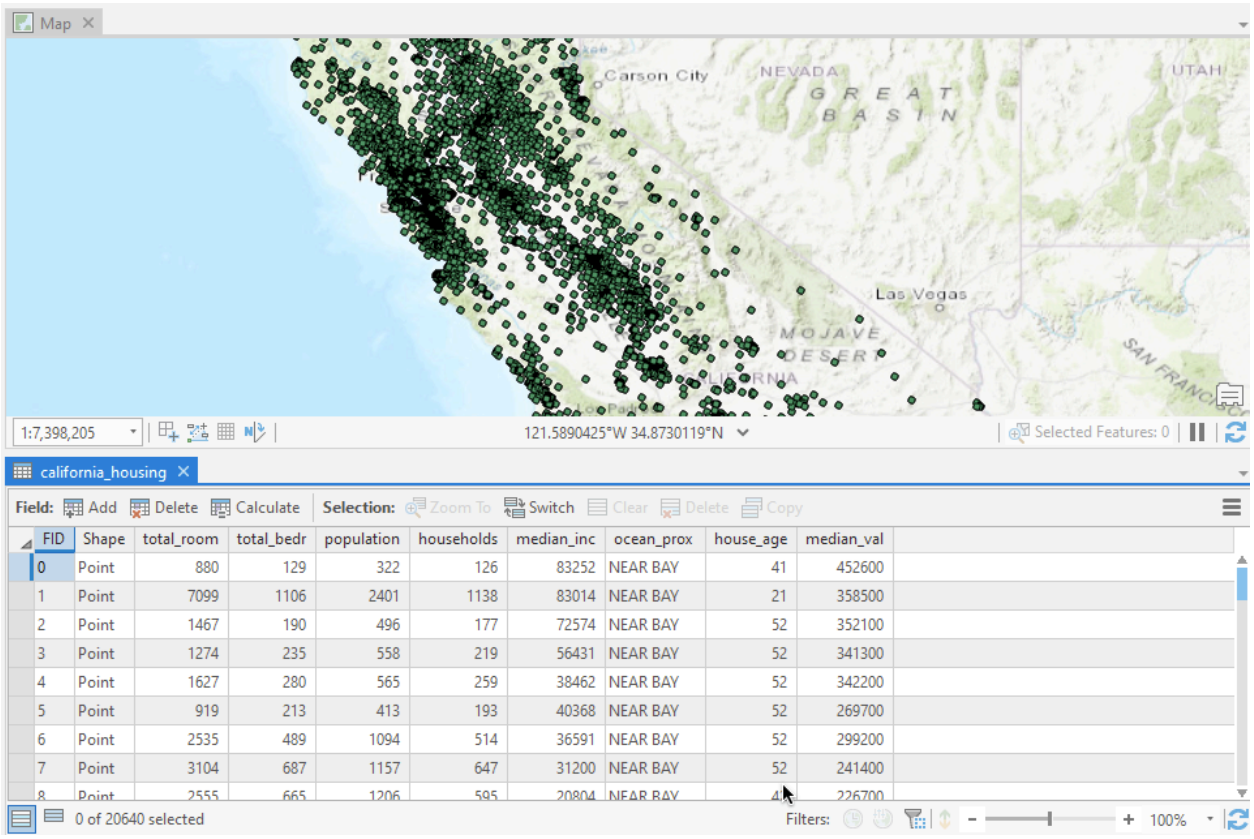
Lab 1: Building a deep neural network model for predicting housing prices at different locations in California

In this lab, you will work with a shapefile dataset that contains median housing price in California at the census block level. This is a dataset that you have worked with in our previous GEO 503 class. Here, we will build a DNN model for making this prediction. To refresh your mind, the shapefile contains the central points of the census blocks, and it also contains some other attributes related to housing in each census block, such as the median house age and the total number of rooms. You can explore the shapefile a bit in a GIS (e.g., ArcGIS).



A bit more information about the shapefile "California_housing.shp": each row contains the data about one census block in California. There are a number of attributes (columns): geometry (i.e., central points of the census blocks), total_rooms, total_bedr (total number of bedroom), population, households, median_inc (median household income), ocean_proximity (the relation between this block and the ocean. Note that this is categorical data!), house_age, and median_val (median house value. This is the value we are trying to predict),).

Your goal is to build a DNN model to predict "median_val". The metric you will use to measure the performance of your model is RMSE:

$$\text{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

Task 1: Load the shapefile data using GeoPandas and preprocess the data (15 pts)

(1) Remove artificial values (5 pts). Using the following code, you can get a quick histogram of all the numeric properties in your data. (Assuming "housing_shp" is the variable name that you use to store the shapefile data; you may need to change it to your own variable name)

housing_shp.hist(bins=50, figsize=(20,15))

From the histogram, you will notice the strange high-value bars for "house_age" and "median_val". This is because these two attributes were capped when the data were recorded. Housing median age will not exceed 52 years while median house value will not exceed $500,001. You can find out the maximum value of house age using:

housing_shp["house_age"].max()

You can then remove these artificial data records using the code below:

housing_shp_cleaned = housing_shp[housing_shp["house_age"] < 52]

In a similar way, you can remove the artificial data records associated with "median_val". After that, if you plot out the histogram again, you should no longer see the strange high-value bars for "house_age" and "median_val".

(2) Prepare dummy variables (5 pts). The "ocean_prox" attribute contains categorical values. In order to use this attribute, we will need to convert it to dummy variables. To see the unique values in "ocean_prox", you can use:

housing_shp_cleaned["ocean_prox"].unique()

To covert the attribute to dummy variables, we can do:

import pandas as pd

housing_shp_cleaned = pd.get_dummies(housing_shp_cleaned)

(3) Extract latitude and longitude as two additional features from the geometry column, and then remove the geometry column (5 pts)

Task 2: Split your data into training (80%) and test (20%) (10 pts)

Tips:

- Set the random_state = 42 when you split the data into training and test
- Don't forget to standardize both your training and test data
- Standardization should not be applied to dummy variables (you will need to first separate your dummy variables, apply standardization on numeric values, and then add back your dummy variables)

Task 3: Build a 3-layer DNN model: The model should have 128, 64, and 1 neurons each layer with relu activation function. Using mean squared error as the loss function and adam as the optimizer. Train the model using the training data with 100 epochs, and set the validation split to 0.2. **(25 pts)**

Task 4: Plot out the epoch-loss curve for training and validation? (5 pts)

Task 5: Use the trained DNN model to predict the test data, and calculate the RMSE of the model. (10 pts)

Task 6: Hyperparameter tuning: Use keras-tuner to try layers from 1-8, and neurons from 16 to 128 with a step of 32. Then, add a final layer of 1 neuron. When performing hyperparameter tuning, set the epochs= 10, max_trials=5, seed=42. After you have identified the best hyperparameters, use them to build a new DNN model **(25 pts)**

Task 7: Result Plotting: Apply your new DNN model to the test data, and plot out the predictions of this new model in a scatter plot with predicted values in y axis and the true values in the x axis. **(10 pts)**

To submit:

- Submit the link to your completed Colab Notebook on UBLearns