# Problem/Motivation/Context

# Proposed solution

## Machine Learning Model

The machine learning model proposed is a Recurrent Neural Network (RNN) that makes use of a Long Short Term Memory Cell. LSTMs have been shown to perform well in learning order dependence in sequence prediction problems (<https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/#:~:text=Long%20Short%2DTerm%20Memory%20(LSTM,complex%20area%20of%20deep%20learning.)>

Since the price for different zipcodes is to be predicted, a separate RNN was trained for each zipcode. Another possible approach is to convert the zipcodes to latitude and longitude using geopy and add lat and long as features to the model. This was not done however.

The RNN consists of an input layer, which takes in a batch of inputs in the form of a time window. The window size is a tunable hyperparameter and is chosen through hyperparameter tuning as explained later. The RNN additionally stores a hidden state in the form of an LSTM Cell. The hidden state is initialised to zero and then changes during training; its purpose is to provide the network with a form of memory. The size of the hidden state is another hyperparameter that must be tuned for the case. The network outputs 1 prediction in its forward path each time a prediction is requested and the prediction is added to the price history to make future predictions based off this prediction.

## Dataset cleaning and preprocessing

The dataset used for the project was the **Zillow Home Value Index (ZHVI). It consists of RegionID, SIzeRank, ZipCode, RegionType, StateName, State, City, Metro, CountyName**

Pandas was used for most of the cleaning in the dataset. The following was done:

1. The ‘State’ column of the dataset was used to filter all other states and keep California states only.
2. All columns other than Zip Codes and Price History columns were dropped.
3. All rows with missing values were dropped. Another possibility was interpolation, but since this is a prototype to demonstrate future potential, the easier approach was taken and we restricted the dataset to zipcodes with all values present.
4. Zipcodes were made the addressing index for convenience

To preprocess the data for training, the price history was broken into windows and MinMaxScaling was done within each window to normalise values between 0 and 1.

# Evaluation

## Dataset splits and training, validation, and testing methodology

An [80, 10, 10] split was used for train, validation and test sets.

**Training: T**he training set was used to train the model using Stochastic Gradient Descent, where one training epoch involves performing gradient descent for each window in the training set. MSE was used as the Loss function as it is a regression task. An adaptive learning rate was used through PyTorch’s Adam optimiser.

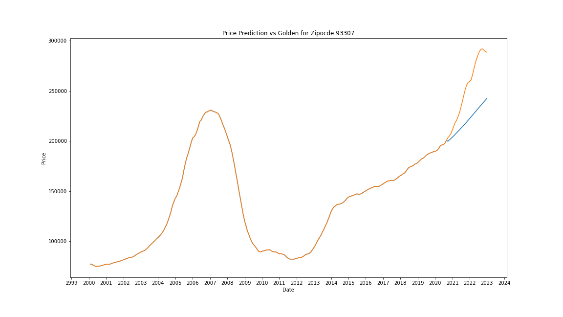
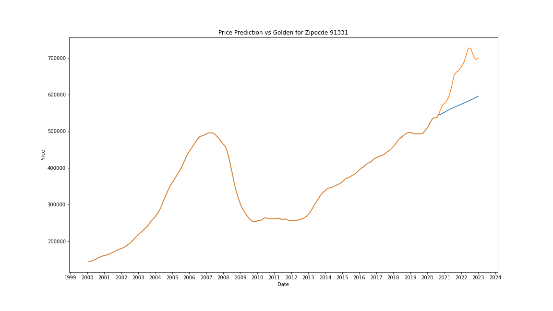
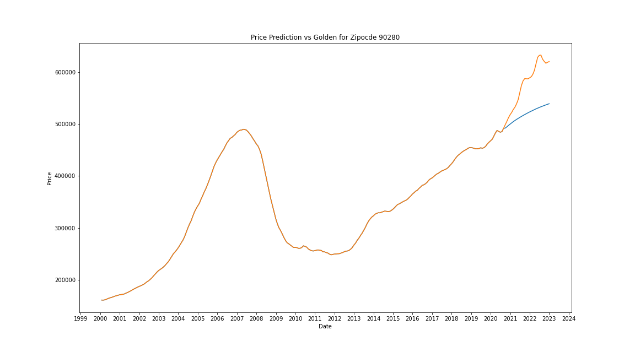
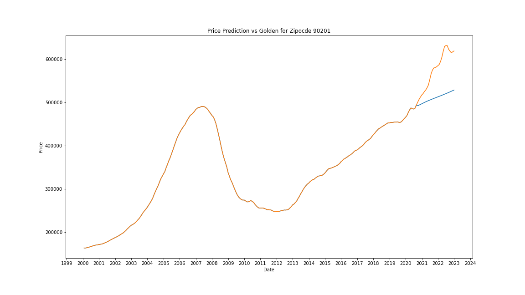
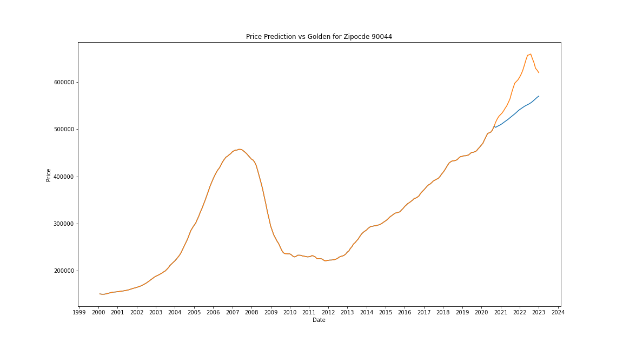
**Validation:** The validation set was used to implement early stopping to avoid overfitting as well to tune hyperparameters like the hidden size for the LSTM Cell as well as the window size for training.

**Test:** RMSE was used as the evaluation metric with the test set since it penalises large errors more than something like MAE, and unlike MSE, a fair comparison can be made between the error and the true value.

## Quantitative Evaluation

The test set was used to evaluate a final score using RMSE. For the zipcodes, the average RMSE on the test set was 66309.8146. For each individual zipcode, it was about 10-15% of the true price.

**ML Models Plots:**

****

**\*\*EVALUATION OF STATISTICAL MODEL (ARTEM AND PANOS) \*\***

## Qualitative Evaluation

**Demo code ready**

## Future Work or Reflection

The RNN model showed a validation RMSE as low as 7500. However, these results did not carry through to the test set. Moreover, the best performing models had window sizes and hidden sizes as large as 100, which relative to the dataset of 276 values is quite large. This implies that the dataset may have been too small. A better model can be trained by using longer or more dense price data.