**US housing Market Analysis**

**Selected Topic**

**Analyzing the housing market in various US counties.**

* Using a variety of tools, we will look at how different factors may contribute the housing prices.

**Reason for Selecting Topic**

* We want to build a model that will predict the change in the price of the houses from 2018 to 2022, how COVID-19 has affected the housing prices and how much equity can be built if the house is purchased now.
* This information can be used by new buyers and people who would like to invest in real estate to know their future price of the house bought or how much price should they expect if they are planning to purchase it after some time.

**Description of the Source Data**

* The data has been sourced from the following location. It shows the prices of houses over the period of 4 years for different zip codes of the US.

**Questions we hope to answer with the data**

* Based on the user input i.e., location, price range and size of the property, the best matching results are populated.
* It will also predict the change in housing prices in the coming one year
* Show the equity that can be built over the house in one year
* Concentration of the COVID-19 affected areas

**Communication Protocols**

* In order to keep updated on the status of each of our parts of the project, we message each other regularly through Slack and organized regular zoom meetings.

**Tools**

* Creating Database
  + PostgreSQL
  + Amazon Web Services (AWS)
* Connecting to Database
  + Psycopg2
* Analyzing Data
  + Pandas
* Machine Learning
  + Imbalanced-learn
  + Scikit-Learn
  + Tensorflow
* Dashboard
  + Tableau
  + Javascript
  + Flask
  + HTML
  + CSS
  + Heroku

**EDA**

* For the housing csv, the rows with null values are dropped
* The **month\_date\_yyyymm** column datatype was changed from integer to datetime.
* The heading was changed to **Date** and column rearranged.
* For the covid cases csv, we changed the datatype of submission date to datetime and replaced the NaN with 0s
* For the population csv, the zip column was converted to 5 digits with int datatype.
* We used third party framework sweetviz to generate reports and the graphical values.

**Machine Learning Model**

* The preliminary data includes columns that describe the environment for each crash that took place in Austin, TX. These features include the weather condition, crash severity, day of the week, vehicle make and model, etc.
* An ERD showcasing the inter-relationships between each of the features from the different datasets can be found [here](https://github.com/cedoula/Final_Project/blob/Deliverable2/QuickDBD-car_crash.png?raw=true).
* After connecting to the database, we printed out the header for each column to see all of the features available. From that list, we chose the features that we believed would have the highest correlation with crash severity.
* The data was split into training and test data using the train\_test\_split function. We used the default 75% to 25% split.
* After careful analyzing, it was determined that the linear models only yielded about 50% correlation. Altering the parameters, such as increasing max iterations and n\_jobs, to these did not increase the accuracy. Neural network model was then used to see if it would have a higher accuracy rate. After adding 8 layers (using Relu, Swish and Sigmoid), the accuracy rate was still at 54%, with 69% loss. This means our model could only accurately predict the outcome of the severity of a crash about 50% of the time.
* We decided to use the decision tree model for our machine learning model. We grouped our crash severity data into two categories, 0 - no injury, and 1 - injury. The benefit of this model is that it can be used to predict our binary outcome. The downside of this model is that if we choose to group our crash severity data differently (the data is grouped into 5 classifications: no injury, possible injury, non-incapacitating injury, severe injury, and fatal injury), we will not be able to use the decision tree model.

**Presentation**

**Dashboard**