Welcome to the NJRealtor-Scrapper wiki!

The Problem:

As a real estate agent in the state of NJ, an investor and a data analyst, I'm highly interested in the various amounts of sales data this market produces for residential, multi-family and land properties. This sales data tells a story; from an investment and agent standpoint, it allows me to understand pricing and sales trends of the whole state, give better insight into where buyers want to be, how much inventory are in theses municipalities, and what percentage of the listing price home sellers are receiving. However, from an analyst standpoint, this data feeds my pure curiosity of how real estate works through numbers and give answers to the questions I have that couldn’t be answered otherwise by intuition or traveling to each city.

Fortunately, as an agent, the New Jersey Association of Realtors (<https://www.njrealtor.com>) aggregate these market statistics and provide them in individual pdfs for all 21 counties and 564 municipalities in New Jersey. Ideally, sifting through the data for a few target municipalities every month wouldn't be an issue; after a few clicks I'd have all the information I need to determine how some local markets are operating and how to make decisions in those markets going forward. However, because I'm interested in the total market's data and keeping track of it on a on a monthly basis, I needed to find a way to properly collect this data to analyze different market statistics to make data-driven decisions.

(Insert section here about city classification types)

TLDR:

* The desirability of a county in NJ can be determined by the respective county’s Total Sales per month and the percentage of the state’s sales per month attributed to that county
* A municipality’s desirability in NJ can be determined by New Listings, Closed Sales, Inventory of Homes for Sale, Total County’s Sales Attributed to that Municipality, and Total State’s Sales Attributed to that Municipality
* Median Sales Prices has no correlation to a municipality’s desirability
* Out of the 21 counties in NJ, approximately 62% (13) of them have ever been considered a desirable county to live in (since 2019)
* Top 5 Most Consistently Desired Counties (in order): Bergen, Monmouth, Morris, Burlington, Camden
* The following counties have never been considered desirable: Salem, Hunterdon, Cape May, Sussex, Cumberland, Hudson, Mercer, Warren
* The Top 5 Counties which are homes to the “Most Desirable” cities (in order): Ocean, Middlesex, Camden, Essex, Monmouth
* Insert List of the Top Most Desired Cities to Live in
* Since 2019, Q3 produces the most sales volume statewide
* Since 2019, the month of October has produced the most sales volume statewide
* Machine Learning models used: Logistic Classification, k-Nearest Neighbor, Decision Tree, Histogram-Based Gradient Boosted Classifier
* Lowest performing Classifier was Logistic Classifier with an F-1 Score of 91% and the highest was the Histogram-Based Gradient Boosted Classifier with an F-1 Score of 98%

The Objective(s):

* Create a process using Python which can automatically log in to my agent portal, download and scrape each pdf for all 564 municipalities
* Clean and store the data then organize the residual pdfs into folders according to their respective counties and municipalities
* Store data in Excel and PostgreSQL databases for further analysis
* Determine the median county's contribution to the state's overall closed sales, the counties which contribute the most to New Jersey's overall closed sales and use the municipalities contribution percentage to that county’s and state overall sales to classify if it’s a ‘Prime’, ‘Productive’ or ‘Marginal’ city.
* Use data to build a ML model that can classify city quality using metrics such as New Listings, Closed Sales, and sales contribution to the county and state.

The Process:

General Purpose:

* Use object-oriented programming (OOP) to create a class to house the functions needed to run the whole process
* Create a logger decorator to inject a logger function into any function where needed
* Create a main function decorator to make sure the process only runs when new data is available on the agent portal
* Use the shelve module to store a process run log, latest data available, run time, run date and the number of days between each data release
* Use the zipfile module to create a zip file containing all the year's pdf files after the December data is extracted

Data Extraction:

* Use Session.Requests to maintain access to the agent portal to download files. Without it, the program will emit an error
* Use BeautifulSoup to confirm when new market data is available on the agent portal
* Use the Requests module to properly parse, create and request the http web address where each individual municipality pdf
  + Create a function which looks for an impromptu checkpoint in the current pdf file's directory which allows the download process to start again at the next file in the event a program error is encountered
* Create a new pdf file and stream the binary data into the new file
  + The available data spanned all the way back to September 2019. The very first run of this process would gather four years’ worth of data. While creating this process, I realized that there was a possibility for the pdf files to be corrupted. The corruption showed up as having last month's data being saved as being collected for the current month or the wrong county or municipality is saved in the pdf
  + If pdf is considered corrupted, store pdf name in a list variable for later processing
* Create a function using PyPDF2 to standardize the process opening, reading and extracting the contents of a pdf and store in a dictionary variable
  + Contents extracted were the Municipality, County, Year, Closed Sales, Days on the Market, Inventory, Median Sales, New Listings, Percentage of Listing Price Received, Months of Supply and the YoY percentage change for each category
  + Assert property datatypes for each respective variable so minimal data cleaning is needed when converting to a pandas dataframe
* Use an assert statement to make sure the month and year of the file name are the same values found in the pdf contents
* Create a function which redundantly checks the current pdf's variable values against previously stored vectors (rows) of the dictionary. If current pdf is found to be corrupted, the file name is stored in a list for later processing and all associated variables set to 0 before being stored in the dictionary

Data Cleaning:

* Use the pandas module to convert the Python dictionary into a dataframe
* Combine the 'Month' and 'Years' column in order to create a datetime pandas series for future time series analysis
* Divide the 'Percent of Listing Price Received' column by 100 to convert all values into percentages
* Filter the dataframe to keep all rows which do not equal 'N.A'
* Insert section about most of the data being right-skewed. Needing to apply a log-transformation to make the data more linear
* Insert section about using StandardScaler() to normalize the data
* Insert section about creating functions using ColumnTransfer() and make\_pipeline() to apply transformations consistently

Data Storage:

* Use pandas to save the dataframe into an Excel file for further analysis
* Use pandas to save dataframe into PostgreSQL database for further analysis and data querying
  + Create connections to PostgreSQL using SQLAlchemy and Psycopg2
  + Psycopg2 would be used to create the database table if it doesn't already exist
  + Pandas and SQLAlchemy will be used to store the dataframe in PostgreSQL and return SQL queries into pandas dataframes

Exploratory Data Analysis (EDA):

* What is the Median Sales Price of the hottest/coldest cities?
* Which counties have the hottest/coldest cities?
* Which qaurter produces the most sales of all time?
* What is the median attributed state sales by county
* What is the median attributed county sales by city
* What is the median attributed county sales by city
* For the latest year and quarter, see what the Top 5 hottest/coldest cities are and the numerical change between this quarter and last quarter

Machine Learning:

* Insert section about model selection
* Insert section about parameter turning, train/test/split
* Insert section about all results

Data Visualization:

* Create a function which uses Matplotlib to create line graphs for every city, grouped by its county for each tracked category
* Create a function which uses geopandas to create a choropleth map for NJ and display the distribution of values across the state
* Use Seaborn to create Pairplots to determine if there are any noticeable trends in the data between each category
* Insert sections for each confusion matrix
* Insert the decision tree plot

Business Impact:

* Insert section here about how my findings can impact my real estate agent and investing business

Future Analysis:

* Use US Census Bureau data (population, median income, income per capita, etc) in combination of the municipality Wikipedia pages to create an RNN to prediction population and median home price trends
* Use NLP to do a text analysis of the WikiPages and use unsupervised ML techniques to do an association analysis of all the city classification types
* Insert section about wanting to be more granular about the Desirability Index and adding more labels
* Insert section about wanting to construct functions which can display the ROC-AUC for multi-class estimators
* Insert section about municipality’s desirability being more nuanced than just the amount of sales. It indicates the amount of desirability but doesn’t explain why. Future projects will hope to explain that