

# NostraDomicile



By:

Jeremy Hutton, Christian Simaan,  
Richard Andrews, Ochaun Marshall

# REDFIN



# Approach

- **Problem:**

- It can be difficult to determine whether a house will sell and what factors will influence its desirability in a particular area.

- **Solution:**

- Create a web application which uses machine learning and determines if a house will sell based on desirable factors in an area and allows users to explore a suite of data visualizations based on their area of interest.

# Important Links

- Website:

- [www.NostraDomicile.com](http://www.NostraDomicile.com)

- Github:

- [www.github.com/nreader72/NostraDomicile](https://www.github.com/nreader72/NostraDomicile)

# Overview

- Functional Requirements
  - User-interface Requirements
  - Security Requirements
  - System Model
  - Subsystems
    - Front-end
    - Back-end
    - Database
    - Machine Learning
  - Demonstration
  - Future Efforts
-

# Functional Requirements

- Users able to input attributes and location for predictive home sale analysis
- Users able to view data visualizations for housing data based on zip code
- Users able to view most influential factors in home sales for a given area
- “About” section with detailed explanation of web application functions and its goal.
- “Blog” section with articles by experts in real estate.
- “Contact” section capable of taking in user feedback via email
- “Help” section which offers users a tutorial

# User Interface and Usability Requirements

- Text area for target zip code in order to return predictions and data visualizations.
- Text areas and drop down boxes for users to enter their home's attributes.
- Submit buttons to return prediction on home sale, most important factors for home sales, and data visualizations..
- The application will load within 1-2 second interval.
- All buttons will conform to the same style.
- Any text area, checkbox, or dropdown box will have helpful instructions.
- Any function of the web application may be reached within 2-3 clicks.
- Any subsequent page within the application will adhere to the same style

# System Interface Requirements

- Application must be successfully hosted and displayed by cloud service(AWS).
- Front end of web application must successfully query database upon user request.
- Database must successfully return requested data run through machine learning algorithm, statistical analysis and data visualization program and front end must successfully display request.



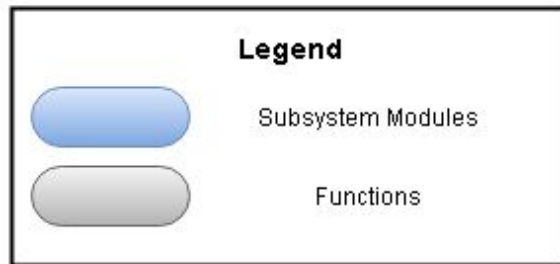
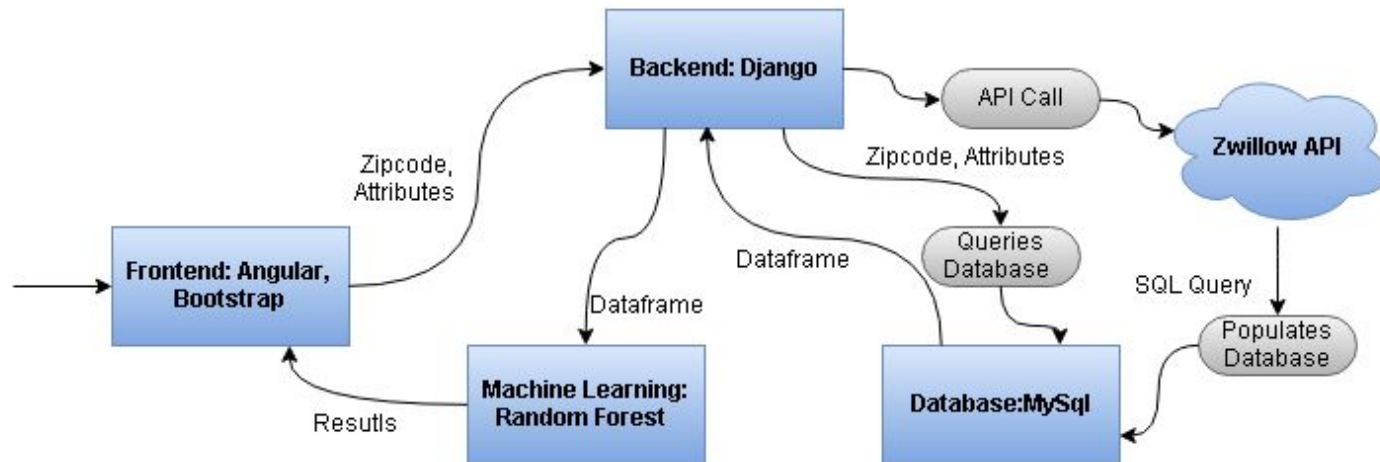
# Security Requirements

- Home data in DB can't be altered except by authorized automated scripts or administrators.
- Realtor blog articles can't be placed, removed or altered except by administrators.

# System Model

- Frontend handles both displaying information to the user, and getting data from the user that will allow our application to analyze attributes.
- The backend analyzes the data that the frontend gets, and send it to the machine learning functions/system. The backend will also make request to the zillow api or our database depending on the request made, and whether we have the data required.
- We use the Random Forest Classifier to analyze the data it gets from the backend, and provide answers to the backend.
- The database will store information we get from requests to the Zillow API, as well as any other information we need to store.

# System Model



# Front End Subsystem

- Goal of the Front End Subsystem:
  - Fulfill the needs set forth in the Functional, User Interface, and Usability Requirements
- Developed using JavaScript and HTML/CSS through the frameworks AngularJS v.1.3.14 and Bootstrap v.3.3.7
- Single Page Web Application
  - Fulfills requirement that any part of the application can be reached within 2-3 clicks
  - Lends to a simple user interface

# Smart Solutions For Data Driven Real Estate Queries



# Front End Subsystem

## Navigation Bar

- Fulfills functional requirements that the application has About, Expert Blog, Contact Us, and Help Sections
- Makes use of Angular Routing to ensure that NostraDomicile.com is a single page application, with no need for page reloads



Please Enter Your Zip Code Of Interest 

Zip Code

I also want to see if my house will sell

Please Enter Your Home's Attributes 

Price

Square Footage

Acreage

Year Built

Neighborhood

School District

Bedrooms ▾

Bathrooms ▾

Story ▾

Type ▾

Parking ▾

Submit Information



# Front End Subsystem

## User Input

- User Interface and Usability Requirements are satisfied by creating text input areas, drop down boxes and submit buttons follow the same design
- For longer load times a loading wheel has been created







### Prediction on Sale

Click the button below to see if your house will sell based on the attributes you have provided.

**Will My House Sell?**



### Prediction on Most Important Attributes

Click the button below to see the most attractive factors leading to home sales in the chosen area.

**Most Important Attributes**



### Data Visualizations

Click the button below to see a collection of valuable statistics pertaining to the chosen area.

**Data Visualizations**

# Front End Subsystem

## Function Cards

- Satisfies the Functional Requirements:
    - Users able to input attributes and location for predictive home sale analysis
    - Users able to view data visualizations for housing data based on zip code
    - Users able to view most influential factors in home sales for a given area
  - Results displayed in pop-up modals which continue the adherence to Single Page Web
- 
- Application

# Back End Subsystem

- Programming language: Python
- Web Framework: Django
- Web Server: Apache
- Host: AWS
- Description: The Django backend will also be broken into many different functions, but it will follow a MVC format. Functions such as making calls to the Zillow API, querying the database, sending data to the frontend to be displayed, etc.

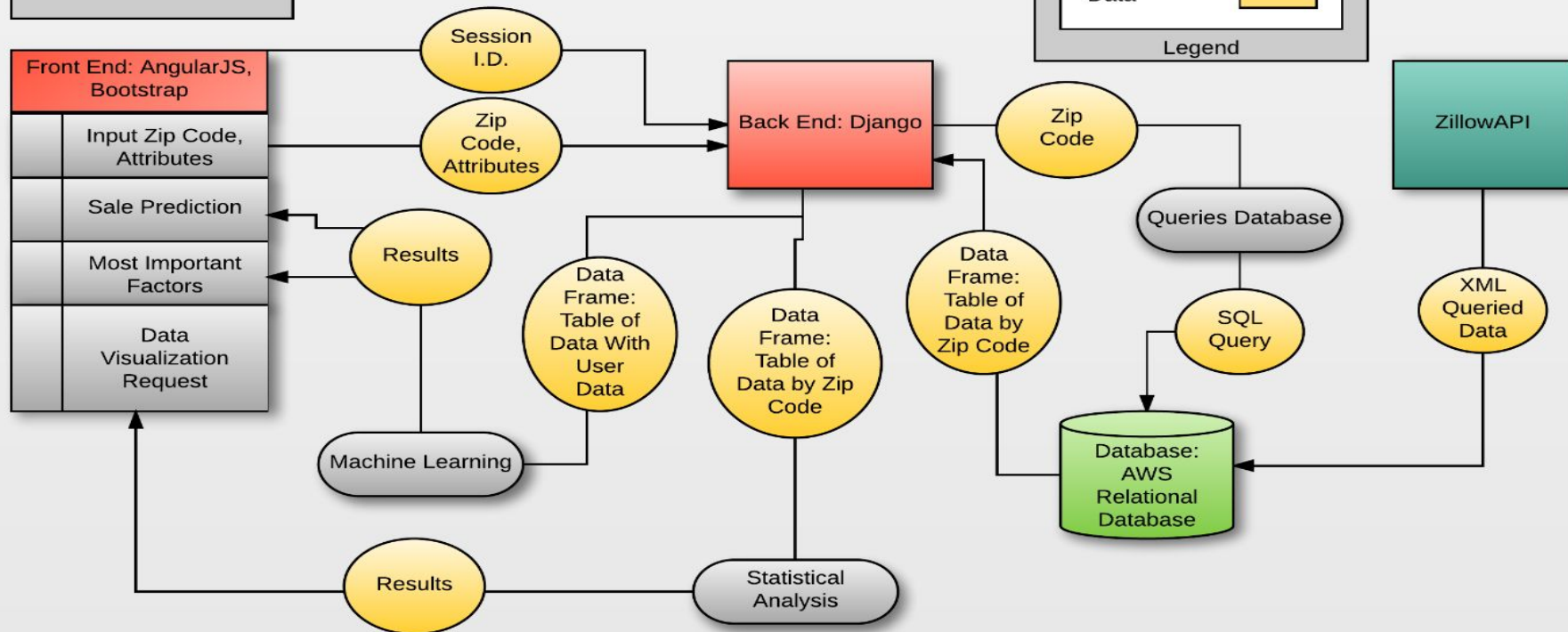
# Database Subsystem

- MySQL Database - data stored in relational DB. Includes, among other things:
  - Zipcode
  - Status (sold/unsold)
  - Home attributes (floor type, parking, etc.)
- Data from Zillow (via PyZillow API).
  - Ideally, data would be directly from MLS database- constantly updated and more complete.

# Machine Learning Subsystem

- Machine Learning with Random Forest.
  - Binary classifications on categorical features is more straightforward
  - Generates an ensemble of decision trees and uses the majority classification of those trees to determine result
  - implemented using Scikit-learn library in Python
  - K-fold Cross validation
- Tasks
  - Generates a trained random forest model from the data from surrounding homes
  - Makes predictions on whether a user's house will sell based on the features of the home
  - Generate a list of features

# NostraDomicile Data Flow Model



# Demonstration

...

# Future Efforts

- Allow multiple data visualization options
- Invite realtors to submit posts to the blog
- Adjust the front-end for mobile phone browsers
- Add support for more classifiers
  - Support Vector Machine
  - Deep Belief Network
- Implement a login system for saving results
- Predicting the price of a home sale by running multiple Random Forests
- Realtor Ranking



# References

1. Breiman, Leo. "Random Forests." *Machine Learning* 45.1 (2001): 5-32. Print.
2. *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. Ijcai. 1995. Stanford, CA. Print.
3. *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference. 2010. van der Voort S, Millman J. Print.
4. Pedregosa, Fabian, et al. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12.Oct (2011): 2825-30. Print.
5. Walt, Stéfan van der, S Chris Colbert, and Gael Varoquaux. "The Numpy Array: A Structure for Efficient Numerical Computation." *Computing in Science & Engineering* 13.2 (2011): 22-30. Print.



*Thanks!*