Problem Set 3: Making Money with ML?
"It's all about location location location!!!"

Due Date: November 7 at 6:00pm in Bloque Neón

1 Introduction

A new start-up dedicated to buying and selling properties just hired you and your team to develop a predictive model. Their objective is to buy the most properties in Cali, Colombia while spending as little as possible.

However, the company doesn't have much information about Cali's housing market. The company has a sample of individual property data on Bogota and Medellin coming from https://www.properati.com.co and aggregate price data shown in Table 1:

Table 1. Prices by City

City	Price
Bogotá	869,755,897
	(899,818,886)
Medellín	639,246,711
	(623,222,205)
Cali	555,314,430
	(601,842,533)

Note: Table shows average prices
and standard deviations in parenthesis.

The data, available in the `dataPS3.zip` file, contains information on listing prices as well as features of the properties. The zipped file includes three other files: training data, testing data, and a submission template.

The company want's to avoid Zillow's fiasco.[1] Zillow developed algorithms to buy houses. However, their models considerably overestimated the price of homes. This overestimation meant losses of about USD 500 million for the company and an approximate reduction of 25% of their workforce.

[1] For more info, see the following article here.

To avoid the fiasco, your team will compete against all other groups, and those with the lowest total budget will get the contract and a bonus.

1.1 General Instructions

The main objective is to construct a predictive model of asking prices. From Rosen's landmark paper ”Hedonic Prices and Implicit Markets: Product Differentiation in Pure Competition” (1974), we know that a vector of its characteristics, $C = (c_1, c_2, \ldots, c_n)$, describes a differentiated good.

In the case of a house, these characteristics may include structural attributes (e.g., number of bedrooms), neighborhood public services (e.g., local school quality), and local environmental amenities (e.g., air quality). Thus, we can write the market price of the house as:

$$P_i = f(c_{i1}, c_{i2}, \ldots, c_{in})$$

However, Rosen's theory doesn't tell us much about the functional form of f. In this problem set, you will explore different models to yield the best prediction possible.

There are two expected outputs:

1. A `.pdf` document.
2. And a `.csv` file with predictions.

The document must contain the following sections:

- Introduction. The introduction briefly states the problem and is an an opportunity to “sell” your predictive model, showing the advantages/disadvantages of your chosen model and expected performance.
- Data[2]. In this problem set, you are required to add expand the variables in your data (remember to expand the training and testing data), at a minimum you have to add seven extra variables:
  - At least 4 predictors coming from external sources; these can be from open street maps.
  - At least 3 predictors coming from the title or description of the properties.

  Treat this section as an opportunity to present a compelling narrative to justify or defend your data choices and help the reader understand your data and its variation. Describe it accordingly with descriptive stats, graphs, etc. At a minimum, you should include:

[2] This section is located here so the reader can understand your work, but it should probably be the last section you write. Why? Because you are going to make data choices in the estimated models. And all variables included in these models should be described here.

  - A table with descriptive statistics
  - Two maps, you can choose what information to show

- Model and Results. When presenting your predictive model include:
  - An explanation of the variables used to train this model, remember to use the variables you added in the previous section.
  - A detailed explanation on how it was trained, the selection of hyper-parameters, and any other relevant information about the model.
  - A discussion of your evaluation measure.

- Conclusions and recommendations. In this section, you briefly state the main take-aways of your work.

2 Additional Guidelines

I expect the following things from the problem set, omission of any of these guidelines will be penalized.

- Turn a `.pdf` document in Bloque Neón. The document should not be longer than 6 (six) pages and include at most 6 (six) exhibits (tables and/or figures). Bibliography and exhibits don't count towards the page limit. You are welcome to add an appendix, but the main document must be self-contained. Specifically, a reader should be able to follow the analysis in the paper and be convinced it is correct and coherent from the main text alone, without consulting the appendix.

- Turn a `.csv` file in Bloque Neón. An example of how the submission file should look like is in the data folder: `submission_template.csv`. This file includes two columns, one with the variable that identifies a property and one with your predicted price. **Do not change the name of the columns**.

- I will judge predictions based on those who spend less money and can buy the most properties. That is if you over-price a property that adds to your tab, if you under-price, you save. However, if the predicted price is under-priced by more that $40mill.$ COP, it will be penalized. In this case, the sale would not take place, i.e., you won't be able to buy the property.

- Please follow the following convention for your `.csv` file name. The file name must include the name `predictions`, followed by your teammates' last names, all separated by underscores, for example, `predictions_gomez_matinez_sarmiento.csv`

- I will assign bonus points based on relative rankings.

- Tables, figures, and writing must be as neat as possible. Label all the variables included. If you have something in your figures or tables, I expect they are addressed in the text.
- The document must include a link to your GitHub Repository.
  - The repository must follow the template.
  - The README should help the reader navigate your repository. A good README helps your project stand out from other projects and is the first file a person sees when they come across your repository. Therefore, this file should be detailed enough to focus on your project and how it does it, but not so long that it loses the reader's attention. For example, Project Awesome has a curated list of interesting READMEs.
  - Include brief instructions to fully replicate the work.
  - The main repository branch should show at least five (5) substantial contributions from each team member.
- The code has to be:
  - Fully reproducible.
  - Readable and include comments. In coding, like in writing, a good coding style is critical. I encourage you to follow the tidyverse style guide.