Madrid, Dec 24th 2018

Master in Business Analytics and Big Data

Machine Learning I

Section O-2

Group H

Use Cases and User Guide

Madrid Real Estate Project

Use cases and User Guides:

As we worked on the project, we thought of two main use cases for this model. One would be from the perspective of an agency listing a new property and the other would be from users looking for good deals as they search for the new apartment.

Real Estate Agency Case:

For a real estate agency, it is important to maximize the rent price but at the same time they want high turnover of properties, so they claim as many commissions as possible and maximize each of them. This can be achieved better using our model as it would predict to them what the price range should be for a property so they maximize the listing price while at the same time minimizing listing time as the price will be better than the competition which is probably optimizing for either price or turnover.

The way this would work would be to deploy the model in a server using Flask creating a REST API which would expect a request with the data used for the model in a specific order and then predicts and sends the response back.

The workflow would be:

1. Agent in the Real Estate company sets up a new property in their database with all required features (it can have more features but not less than the required by the model)
2. The information would be saved in the database of the company and at the same time a script would query our REST API with the required information.
3. The model would then predict the price and then send it back through a response.
4. The same script that selected the features from the database and made the query would then update the database with this new predicted listing price and a range based on the error of the model.
5. The UI for the Real Estate company would show these metrics and helps the agent select the optimal price for the property

Users finding apartments Case:

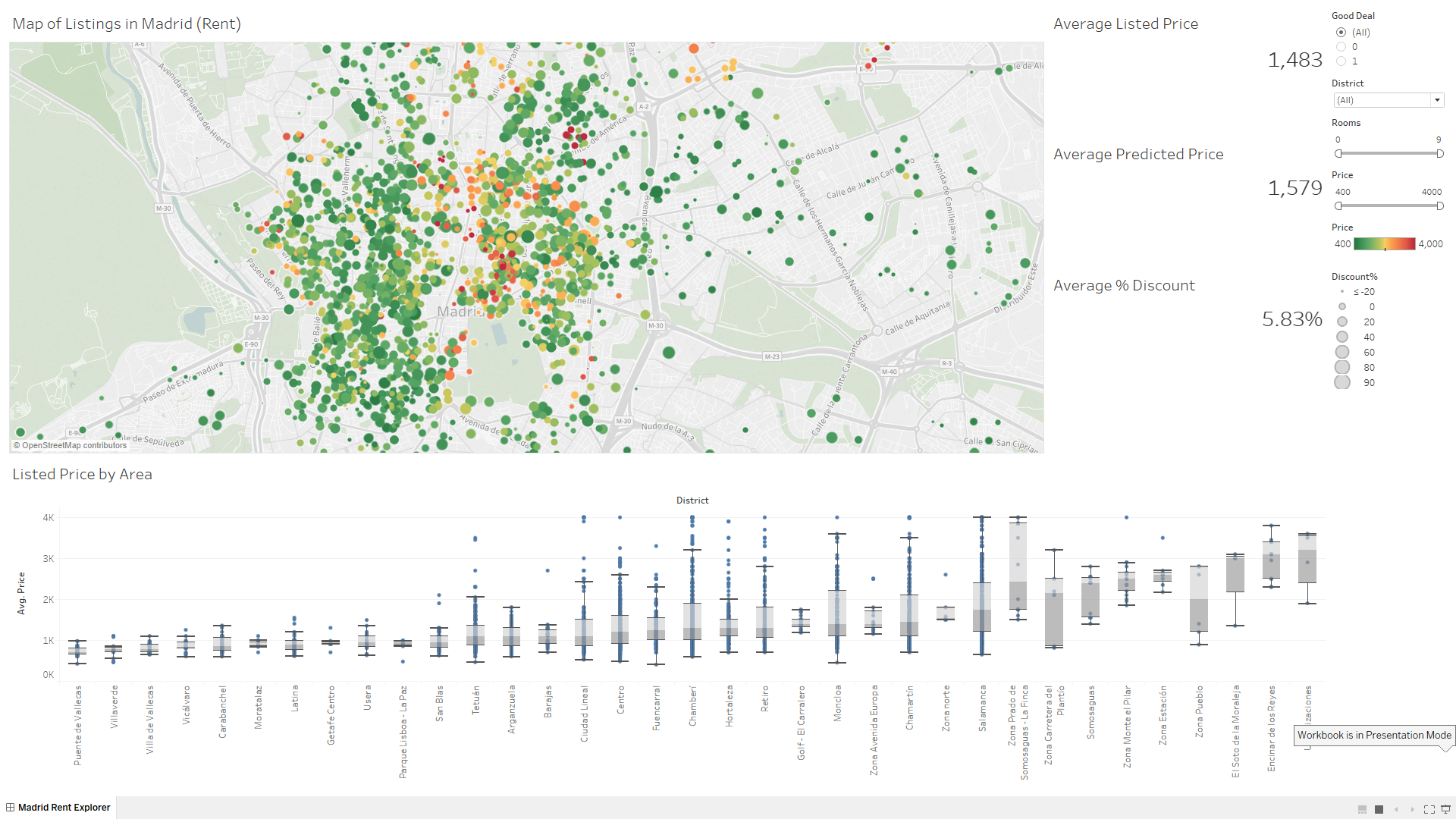
From the other side, this model would benefit users trying to find new properties to rent as they could explore, filter and find the places that fit their needs and are good deals.

However, as a user doesn’t deal with case by case needing predictions. The approach needs to be completely different.

In this case we deploy the model in a script in a server. Every day the script would query the information from Idealista, updating the listings in the database and removing the listings that are not there anymore from the active database and adding the new listings. These new and updated listings go through our model and script calculating the benefit and the %Discount and adding them as attributes in the database.

This database will be connected to a frontend solution that lets a user visualize, filter, and search for their respective listing.

For this case we decided to make a static Proof of concept using Tableau. Which can be seen here:



Here we can see a dashboard with every listing in our data set. Colored by the price and sized by the amount of discount. The bigger the dot, the better the discount between predicted/listed price. And the color is dynamic to show the low/high range of the selection.

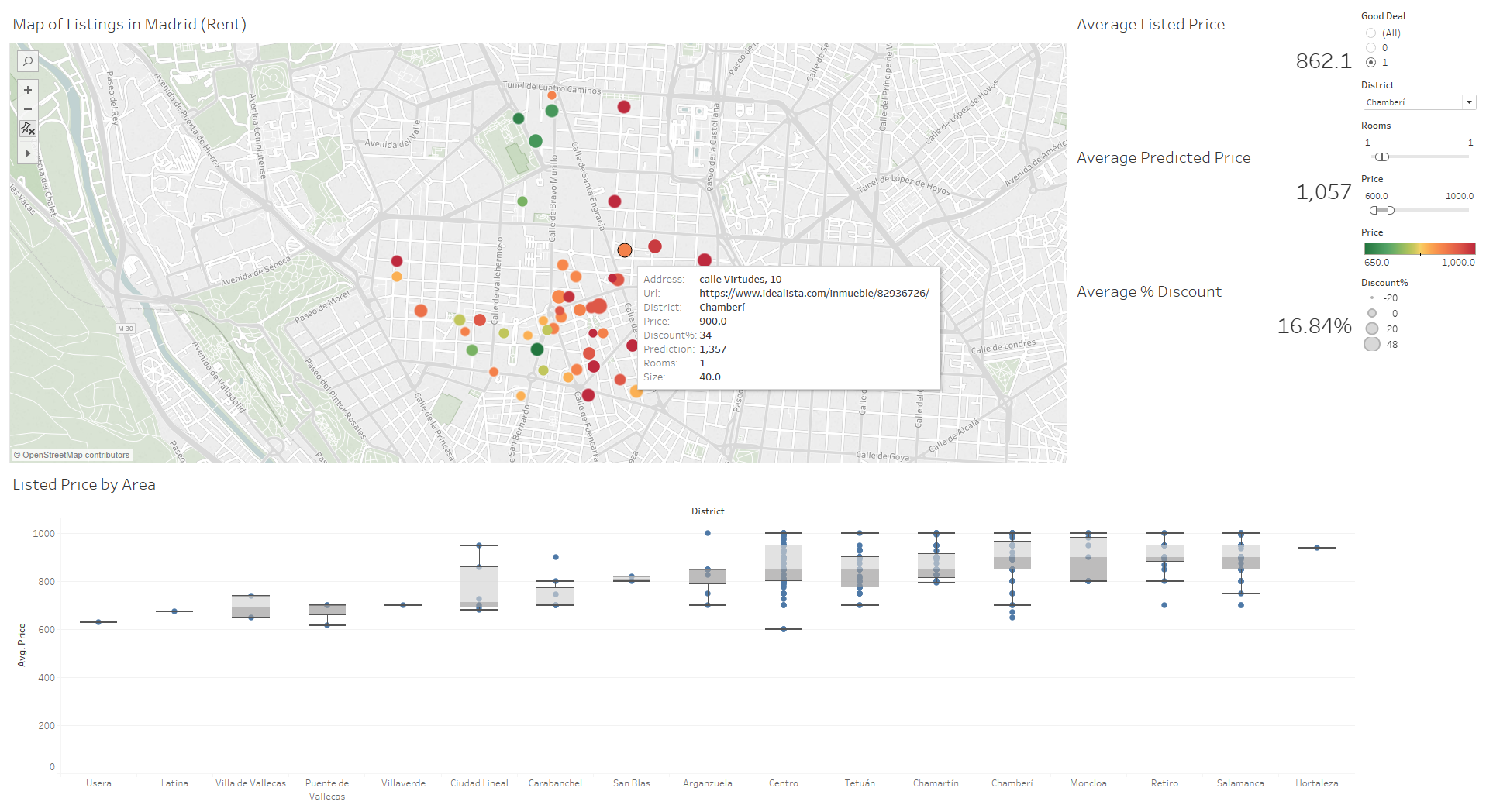
At the bottom we can see boxplots of listed prices by Area, so a customer can find an area that fits their budget much faster.

The user can also filter by District, Rooms, whether the listing is a good deal or not, and the price range.

On the right we see the average listing price, predicted price and discount for the selection. So if a user has a couple properties in mind, he can check those prices against the average in the zone or the average between his own selections.

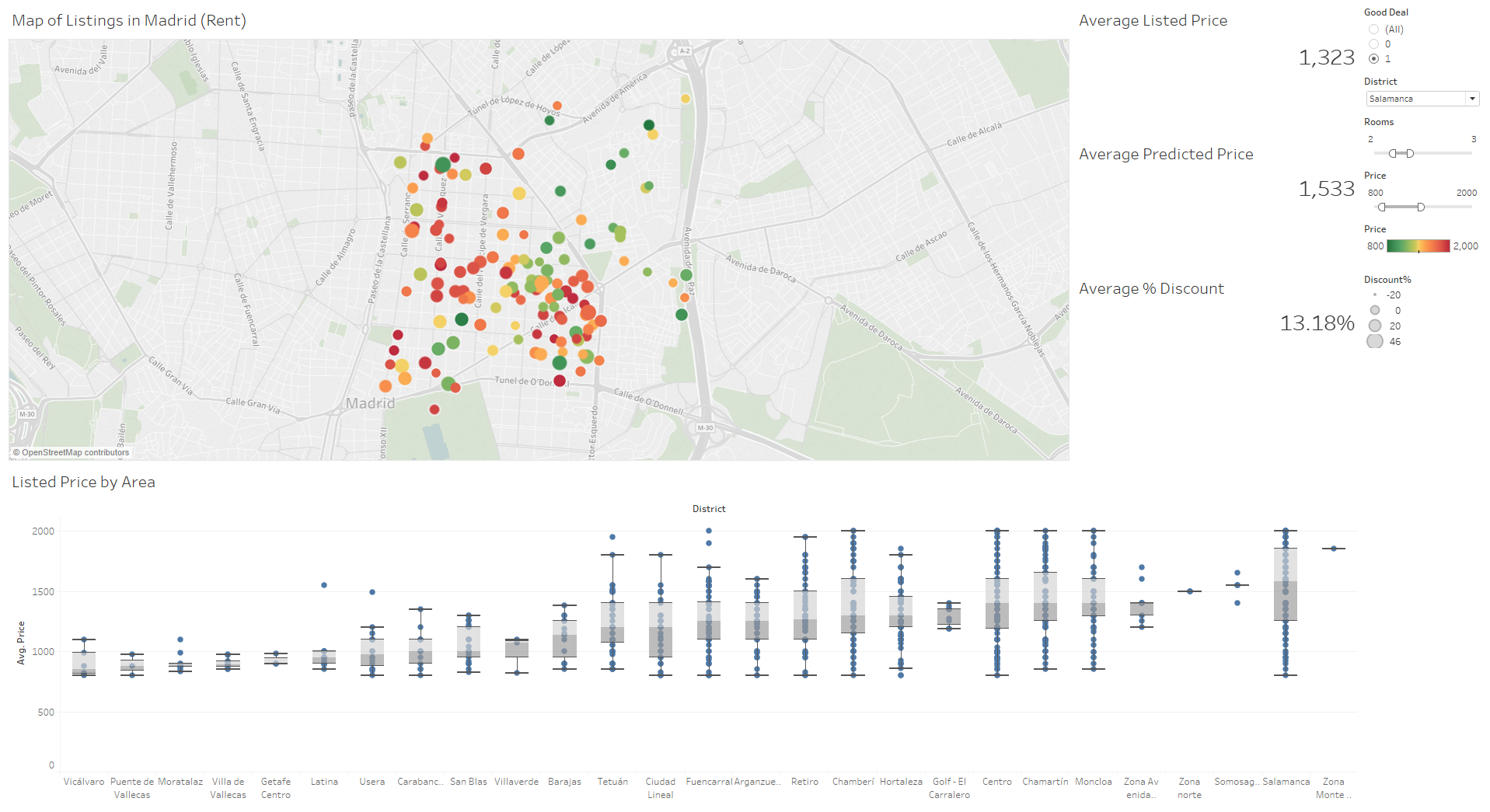
For example, let’s say we have 2 users. One is looking for a 1-bedroom apartment in Chamberi between 600 and 1000 Euros and that it’s a good deal. And the other is looking for a 2- or 3-bedroom apartment in Salamanca with an broader budget between 800 and 2000 Euros but he wants bang for his buck so he wants only Good Deals

User 1 would see:



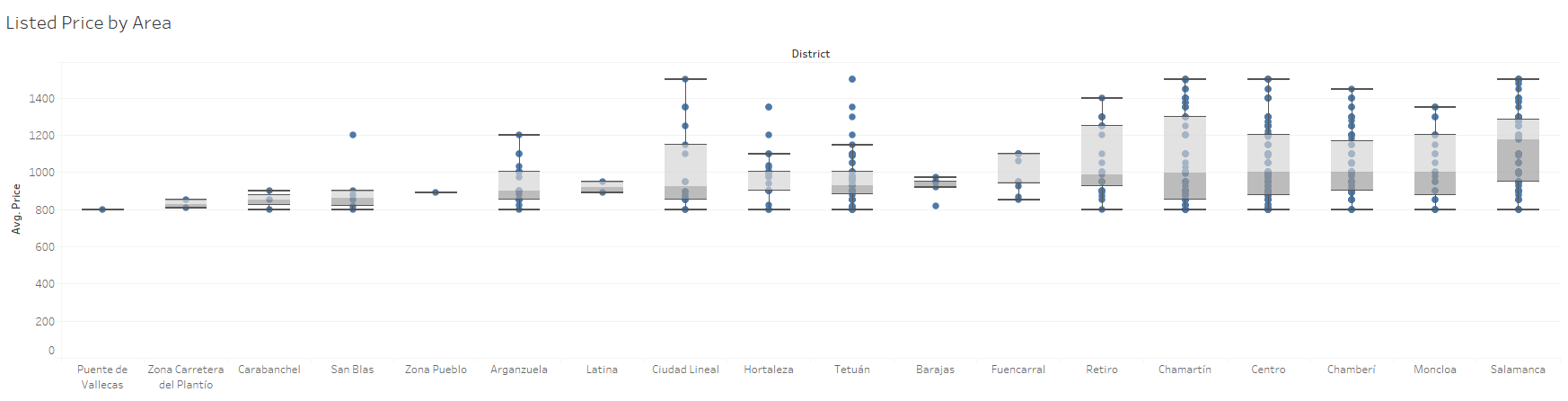
Which shows the available apartments and their average listing/predicted/discount. And he can then hover to see information about each listing.

User 2 would see:

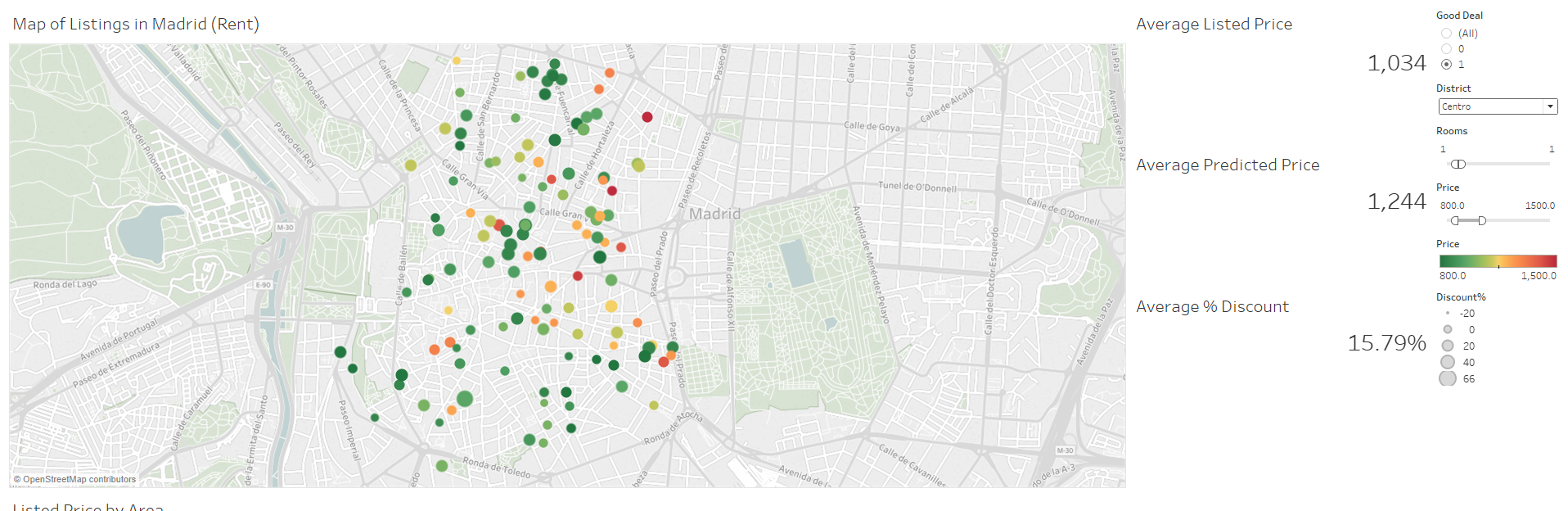


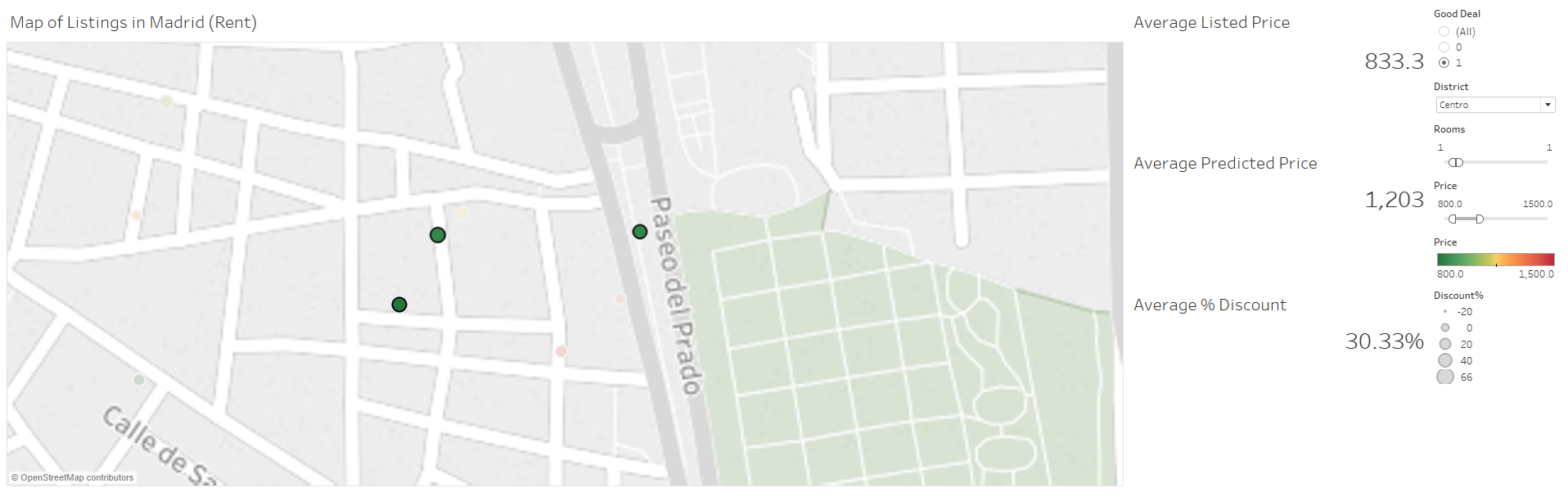
Which shows all the apartments that the listing price is less than the predicted price and that meets his criteria. This helps them make better informed decision.

As a third use case. We though of someone like a Student or Expat that might be moving to Madrid so he doesn’t even know which area to look for. He knows he needs 1 bedroom and his budget is between 800 and 1500 Euros. He would filter by those variables and see the bottom graph



Which gives him some information like that Salamanca is on average more expensive than Moncloa. But as he wants to be near the Retiro Park, he hovers to find those properties West of Retiro to find they are in the “Centro” district. So, he then filters the good deals in that area by his criteria and finds:



As he wants to be near the park, he finds that cluster of green bubbles closest to it and selects them to find: 

That those 3 properties have an average listing price of 833 Euros and their average predicted price is 1203 Euros for an average discount of 30.33%. Meaning that he should check those properties.

This model is then retrained every month to take into account seasonality and trends in the market to keep it as updated as possible. This allows our users to have access to the most accurate data possible.